Formalization of Discrete-time Markov Chains in HOL

Li Ya Liu

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

May 2013

CONCORDIA UNIVERSITY

Division of Graduate Studies

This is to certify that the thesis prepared

By: **Li Ya Liu**

Entitled: **Formalization of Discrete-time Markov Chains in HOL**

and submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Dr. Mamoun Medraj (Chair)

_____ Dr. Prakash Panangaden

_____ Dr. Olga Ormandjieva

_____ Dr. Dongyu Qiu

_____ Dr. Samar Abdi

_____ Dr. Sofiène Tahar

Approved by _____

Chair of the ECE Department

_____ 2013 _____

Dean of Engineering

ABSTRACT

Formalization of Discrete-time Markov Chains in HOL

Li Ya Liu

Concordia University, 2013

Markov chains are extensively used in the modeling and analysis of engineering and scientific problems which can be expressed as random processes with the memoryless property. Usually, paper-and-pencil proofs, simulation or computer algebra software are used to analyze Markovian models. However, these techniques either are not scalable or do not guarantee accurate results, which are vital in safety-critical systems. To improve the accuracy of the analysis, probabilistic model checking has been recently proposed to formally analyze Markovian systems. However, model checking suffers from the inherent state-explosion problem and thus has a very limited scope in terms of analyzing Markovian models.

In order to overcome the above mentioned limitations, this thesis advocates the usage of higher-order-logic theorem proving for conducting the analysis of Markov chains. We present the higher-order-logic formalization of Discrete-time Markov Chains with finite number of discrete states. We also verify some of their most widely used properties using a theorem prover. These foundations allow us to formally express and reason about Markov chains within the sound core of a theorem prover and thus attain precise results. Moreover, by building upon these foundational results, this thesis also presents the formalization of classified discrete-time Markov chains and hidden Markov chains in higher-order logic. These are widely used concepts in the analysis of Markovian models and thus allow us to tackle the formal analysis of a

wide range of engineering and scientific systems. For illustration purposes, the thesis also presents some applications including a binary communication channel, the automatic mail quality measurement (AMQM) protocol, a DNA sequence, a least recently used (LRU) stack model and the birth-death process.

To My Husband and Son, My Parents and My Brother

ACKNOWLEDGEMENTS

I would like to deeply thank Dr. Sofiène Tahar for all his help, guidance and support during my Ph.D studies. His encouragement played a vital important role in my research progress, it will even inspire me in my future life. I would also acknowledge his efforts for providing a friendly group – Hardware Verification Group (HVG), where I spent the most important four years of my life. Many thanks to Dr. Osman Hasan for his patient assistance and professional advice throughout my four-year study and research, where he has been practically my second Ph.D supervisor. I also sincerely thank Dr. Vincent Aravantinos for his professional advice, insightful criticisms and friendly encouragement with HOL4 and many other aspects of theorem proving. Their inspiration helped me pass through many frustrating moments during my research. I would like to express my gratitude to the members of my thesis committee for their help on each stage of my research project.

I am deeply thankful to my parents, who always show great interest in my research and provide endless love to motivate me to be better and better in my studies and research. I also would like to thank the love of my life, Liu Peng, for his unconditional support and inspiration for me on pursuing doctoral studies. No doubt, my Ph.D dream could have never become true without his support and encouragement. I also want to thank my son, Jiale Martin, who had to endure my busy study process as he has been growing up. He brought me so much happiness in my life that helped me overcome difficult times! Nothing would be possible without their love and support.

I would like to thank Dr. GuanYong and Dr. Jin Shengzhen from the Chinese Academy of Sciences, who provided me lots of support on mathematics when I was struggling with the challenges in my research. Also, I appreciate Mr. Mukesh Kumar

Agrawal from IIT Delhi, who was very helpful during his internship in our HVG.

Last but not least, I would like to thank all my friends in the Hardware Verification Group for their support and motivation, though I do not list all their names here.

TABLE OF CONTENTS

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF ACRONYMS xiii

1 Introduction **1**

1.1 Related Work 4

1.1.1 Simulation 4

1.1.2 Computer Algebra Systems 7

1.1.3 Probabilistic Model Checking 8

1.2 Proposed Methodology 9

1.3 Thesis Contributions 12

1.4 Thesis Organization 14

2 Preliminaries **15**

2.1 HOL Theorem Proving 15

2.2 Probability Theory 19

3 Discrete-time Markov Chain in HOL **23**

3.1 Definition of Discrete-time Markov Chain 23

3.2 Classical DTMC Properties 28

3.2.1 Joint Probability Theorem 28

3.2.2 Chapman-Kolmogorov Equation 29

3.2.3 Absolute Probability 30

3.2.4 Reversibility Property 31

3.3 Stationary Distributions 33

3.4 Stationary Process . . . 36
3.5 Applications . . . 38
3.5.1 Binary Communication Model . . . 39
3.5.2 AMQM Protocol . . . 43
3.6 Summary and Discussion . . . 49

4 Classified Discrete-Time Markov Chain in HOL 52

4.1 Classified States . . . 52
4.2 Classified DTMCs . . . 54
4.3 Long-term Properties . . . 57
4.3.1 Positive Transition Probability . . . 58
4.3.2 Convergence Analysis . . . 62
4.4 Applications . . . 63
4.4.1 LRU Stack Model . . . 63
4.4.2 Discrete-time Birth-Death Process . . . 68
4.5 Summary and Discussion . . . 73

5 Formalization of Hidden Markov Model 76

5.1 Definition of HMM . . . 76
5.2 HMM Properties . . . 79
5.2.1 Joint Probability of HMM . . . 80
5.2.2 Observation Sequence Probability . . . 81
5.2.3 Best Path Selection . . . 82
5.3 Proof Automation . . . 83
5.4 Application: DNA Sequence Analysis . . . 86
5.5 Summary and Discussion . . . 90

6 Conclusions and Future Work **92**

6.1 Conclusions 92

6.2 Future Work 94

Bibliography **98**

Biography **110**

LIST OF TABLES

2.1 HOL Symbols and Functions 18

4.1 Formalization of Classified States 55

LIST OF FIGURES

1.1 Markov Chain Application Fields 2

1.2 Proposed Framework 11

3.1 State Diagram 39

3.2 Channel Diagram 39

3.3 Tag Collection Process 44

3.4 DTMC Model of the AMQM Protocol 45

4.1 LRU Stack Updating Procedure 64

4.2 State Diagram for the LRU Stack Model 65

4.3 State Diagram of Discrete-time Birth-Death Process 68

5.1 5' Splice Site Recognition Model 87

6.1 Future Research Directions 94

LIST OF ACRONYMS

AMQM	Automatic Mail Quality Measurement
AP	Acknowledge Process
CAS	Computer Algebra System
CDF	Cumulative Distribution Function
CP	Command Process
DNA	Deoxyribonucleic acid
DSA	Dynamic Spectrum Access
DTMC	Discrete-Time Markov Chain
GSPNs	Generalized Stochastic Petri Nets
HMM	Hidden Markov Model
HOL	Higher-Order Logic
HOL4	HOL4 Theorem Prover
IID	Independent and Identically Distributed
IPC	International Post Corporation
LRU	Least Recently Used
LP	Listen Period
MCMC	Markov Chain Monte Carlo
MDP	Markov Decision Process
ML	Meta-Language
MRF	Markov Random Field
PCTL	Probabilistic Computer Temporal Logic
PDF	Probability Density Function
PMF	Probability Mass Function

PRISM	Probabilistic Symbolic Model checker
RF	Radio Frequency
RFID	Radio Frequency IDentification
RMS	Reliability, Maintainability and Safty
SML	Standard Meta-Language
SPN	Stochastic Petri Nets
SPNP	Stochastic Petri Nets Pakage
WP	Wakeup Period

Chapter 1

Introduction

In our daily life, most natural phenomena are random or unpredictable. To quantify the possibility of the appearance of random events, probability theory has been built as an important branch of mathematics for probabilistic analysis of the random phenomena. The majority of the randomness has some sort of time-dependency. For example, noise signals vary with time, the duration of a telephone call is somehow related to the time it is made, population growth is time dependant and so is the case with chemical reactions. Thus, various probabilistic models are employed to describe the behaviors of systems. Diverse random processes exhibit the *memoryless property* [8], which means that the future state depends only on the current state and is independent of any past state. In science and engineering domains, numerous applications desire to predict the future states by the given current state and these applications are usually modeled as *Markov chains* [8].

More than one hundred years ago, Andrey Markov, a Russian mathematician, proposed a series of foundations related to random processes that exhibit the *memoryless property* (also called the *Markov property*). These random processes are now

commonly known as *Markov processes* and Andrey Markov's findings are commonly termed as the *Markov Theory*. Markovian systems can be broadly classified as four types based on their time and state parameters: discrete-time and discrete state, discrete-time and continuous state, continuous-time and discrete state, and continuous-time and continuous state. The *discrete-time and discrete state Markov Process* is usually called the *Discrete-Time Markov Chain (DTMC)* [8].

A DTMC model consists of a list of the possible states of the system along with the possible transition paths among these states [88]. This kind of a simple structure of a DTMC model makes it handy to study various behaviors such as transient behavior and limiting behavior, of a stochastic process with discrete spaces by solving the linear equations. Therefore, DTMC is the most widely used stochastic process for analyzing the reliability, maintainability and safety of real-world applications as shown in Figure 1.1. For instance, in biologic science, a typical Markov chain namely, the *Birth-*

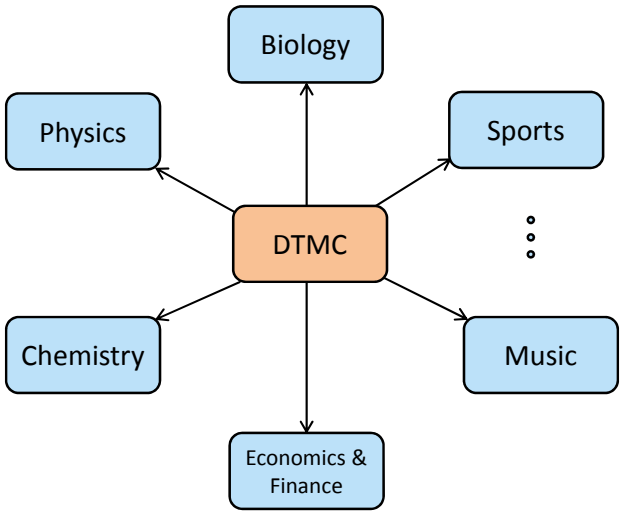

Figure 1.1: Markov Chain Application Fields

Death process [109], is applied in modeling biological populations. Also, the Markov chain theory has been applied in physics (such as thermodynamics [65] and statistical

mechanics [97]). Moreover, the enzyme activity and the growth of copolymers can be modeled as a Markov chain in chemistry [104]. Furthermore, a variety of economics and finance phenomena, such as asset prices [35] and market crashes [93], are described as Markov chains. Some music composition algorithms [54] based on the Markov chain theory are employed in software. In addition, the Markov chain theory has been applied in advanced baseball analysis [19].

A DTMC is further divided into two main categories. It may be *time homogeneous*, which refers to the case where those Markov chains exhibit the constant transition probabilities between the states, or *time inhomogeneous*, where the transition probabilities between the states are not constant and are time dependant. Furthermore, DTMCs are also classified in terms of the characteristics of their state-space. For example, some states can be reached from all other states and some others are those that once entered then cannot be left. In practice, these states are the most attractive states in the dynamic analysis of Markovian systems. Regarding the features of the states in their state space, DTMCs are categorized into different classes, such as *irreducible DTMC*, *aperiodic DTMC*, *absorbing DTMC* etc., where these classified Markov chains [88] are widely used to simplify the analysis of long-term behaviors for most applications.

As an evolved Markov chain, the Hidden Markov Model (HMM) [72] is a stochastic process involving an underlying Markov chain which changes the output of the random functions associated with each state in the Markov chain. The observer can visualize the output of the random function but not the underlying states. That is why the Markov chain involved in this process is called *hidden Markov chain*. The observed sequence is said to be *conditionally independent* [114] on this hidden Markov chain.

Initially, HMMs were proposed to solve optimal linear filtering problems as the simplest dynamic Bayesian networks. However, due to their usefulness in effectively analyzing probability distributions over a sequence of observations, HMMs are now extensively used in a variety of applications involving speech recognition, cryptanalysis, molecular biology, data compression, financial market forecasting and artificial intelligence, as a ubiquitous tool.

1.1 Related Work

Traditionally, engineers have been using paper-and-pencil proof methods to perform probabilistic and statistical analysis of Markov chain systems. Nowadays, real-world systems have become considerably complex and the behaviors of some critical subsystems need to be analyzed accurately. The computations tend to grow tremendously and it becomes practically impossible to analyze a complex system precisely by paper-and-pencil methods due to the risk of human errors. Therefore a variety of computer-based techniques, such as simulation, computer algebra systems and probabilistic model checking have been recently proposed to analyze Markovian models.

1.1.1 Simulation

Simulation is the most commonly used automated technique for analyzing Markovian models. In a simulator, such as the *simulink* toolbox in *Matlab* [107], the automatic analyses are conducted by providing the system model and the input samples. The arbitrary samples generated by traditional random functions cannot ascertain the behavior of desired systems for all possible cases. The *Markov Chain Monte Carlo* (MCMC) method [25] tends to increase the precision of the analysis by using the sampling approach to approximate the desired distribution in terms of the residual

effect of the initial position. Some of the sophisticated MCMC-based algorithms are capable of producing samples matching the given probability distribution, but the major limitation of MCMC is that it generally requires hundreds of thousands of simulations to evaluate the desired probabilistic quantities and becomes impractical when each simulation step involves extensive computations. In order to improve the computation efficiency, some approximations are introduced in the complex analysis process. Especially, in the long-term behavior analysis, the high computational costs are associated with $vp_{ij}^{(n)}$ for large values of n, where v is any probability vector and $p_{ij}^{(n)}$ is the n-step transition probability from state i to state j. Most simulators utilize an equilibrium vector to approximate $vp_{ij}^{(n)}$ in order to reduce the computation time.

Many reliability evaluation software tools integrate simulation and numerical analyzers for modeling and analyzing the reliability, maintainability or safety of systems using Markov methods, which offer simplistic modeling approaches and are more flexible compared to traditional approaches, for example, Fault Tree [18]. Some prevalent tool examples include *Möbius* [84] and *SHARPE* [105]. These tools mainly provide the services on analyzing the failure or repair of a model, which may occur in the lifetime of any product. Some other software tools used for evaluating performance, e.g., *MACOM* [102] and *HYDRA* [85], take the advantage of a popular Markovian process algebra [7], i.e., *PEPA* [92], to model systems and efficiently compute passage time densities and quantities in large-scale Markov chains.

Another technique, *Stochastic Petri Nets* (*SPN*) [32], has been found as a powerful method for modeling large-scale systems because it allows local state modeling instead of global modeling. SPN are utilized to model the stochastic systems and offer the capability of analyzing large and complex models. The Markov chain of an SPN is modeled by means of a reachability graph [61]. The prevailing software

tools of stochastic petri nets are *SPNP* [14] and *GreatSPN* [30]. These tools can model, validate, and evaluate distributed systems and analyze the dynamic events of the models by means of embedded Markov chain theory. For example, the quantitative analysis of Generalized Stochastic Petri Nets (GSPNs) [33] mainly depends on a Markovian solution, in which the models are described as semi-Markov processes in order to calculate the steady state distributions of the stochastic systems. The calculations are based on numerical methods again, which is the main limiting factor of the application of SPN for analyzing safety-critical system models.

Various simulation-based HMM analysis tools, dedicated to a particular system domain, have been reported in the literature. Some prominent examples include *HMMTool* [43] as part of the *NHMMtoolbox* [98] to predict daily rainfall sequence. *ChIP-Seq* [12], MArkov MOdeling Tool (*MAMOT*) [23] and *HMMER* [42] are some of the popular simulation software in biological research. However, as mentioned earlier, due to their approximate nature, all these simulation techniques are not reliable enough for safety-critical applications.

In general, the analysis based on the simulation technique can never be termed as 100% precise due to the inaccurate nature of the underlying algorithms, which are based on numerical methods that generate inaccurate results. In addition, many rounding errors also creep into the analysis due to the involvement of computer arithmetic. Such approximations and inaccuracies introduced by numerical methods pose a serious problem while analyzing highly sensitive and safety-critical applications, such as nuclear reactor controllers and aerospace computing systems.

1.1.2 Computer Algebra Systems

Computer algebra systems (CAS) provide fully automated support for analyzing Markovian models and symbolic representation of Markovian systems using a friendly human-machine interface. The symbolic analysis overcomes the limitations associated with the inaccurate results generated by applying numerical methods. It facilitates the calculus problems, such as multiprecision arithmetic, operations on polynomials and the evaluation of the eigenvalues for linear equations. Recently, the CAS tool *Mathematica* [77] has introduced a Markov chain analysis tool-box which offers a completely automatic analysis. Another well-known CAS, *Maple* [75], also utilizes Markov chains for solving financial problems by automatically constructing transition matrices in Markovian models.

On the surface, CASs provide results as generic formulas and are similar to theorem provers. However, the simplifications performed in the CASs are not strictly mathematical as they are not able to deal with side conditions. For example, *Mathematica* [77] returns 1 as the answer when given "x / x" as the input. It is clear that "x / x = 1" holds only when x $\neq$ 0. Moreover, CAS algorithms cannot simplify some expression. For example, the following simplification rule is not supported by *Mathematica* [37]:

$$\sqrt{x^2} = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x \leq 0 \end{cases}$$

As a result, the core of computer algebra systems cannot be considered sound enough to guarantee the correctness of the final result. Moreover, the presence of huge symbolic manipulation algorithms, which have not been verified in the cores of CASs also make the analysis results untrustworthy. In addition, if the formulas given by

CASs are instantiated with concrete values for their parameters, then the real numbers obtained for the corresponding results are mainly calculated by applying some numerical methods, such as the Jacobi Over-Relaxation, Gauss-Seidel and Successive Over-Relaxation algorithms [6], for the affordable computation reason. In this case, the results include approximations.

1.1.3 Probabilistic Model Checking

Recently, *probabilistic model checking* [100] has been proposed for analyzing Markovian systems by modeling stochastic behaviors using probabilistic state machines and exhaustively reasoning about the probabilistic properties of numerous Markovian systems in a precise logic. Some of the commonly used probabilistic model checkers are *PRISM* [94], *VESTA* [103] and *Ymer* [113]. Among them, PRISM is the most widely used, where the system model is constructed as the state transition graph and the properties are specified using *Probabilistic Computer Temporal Logic* (PCTL).

Probabilistic model checking tools perform the formal analysis of Markov chain models automatically and provide counter-examples if the verification of a system model failed. Although they offer exact solutions, they are drastically limited by the state-space explosion problem [5] and the limited expressiveness of the property specification language. Furthermore, the time for analyzing the properties of a system is largely dependent on the convergence speed of the underlying algorithms. For example, the *Power method* [90], which is a well-known iterative method, is applied to compute the steady-state probabilities (or limiting probabilities) of Markov chains in PRISM. Such algorithms are mainly based on numerical methods, which usually bring about approximations in the results. These approximations reduce the accuracy of the results or even execute unreliable results. Moreover, probabilistic model checking

tools often utilize unverified algorithms and optimization techniques [108]. Finally, model checking cannot be used to verify generic mathematical expressions for probabilistic analysis due to the inherent state-based nature of the approach. To the best of our knowledge, no case study of HMM formal analysis has been reported in the open literature, except for some model checking algorithms based on the *Probabilistic Observation CTL* (POCTL), which is used for modeling and specifying properties of parameterized HMMs. The complexity of these algorithms depends on the size of the model and the number of variables involved in the property formula. This factor, coupled with the inherent nature of model checking, severely limits the usage of this algorithm for analyzing real-world examples.

1.2 Proposed Methodology

With an extensive usage of Markov chains in modeling and analysis of systems, the availability of their accurate analysis techniques has become imperative. Various techniques have been proposed for analyzing Markovian models as described above, but none of them can guarantee providing accurate analysis for all sorts of Markovian models. Probabilistic model checking provides a formal analysis of various Markov chain models, but to the best of our knowledge it cannot cater for some, for example hidden Markovian models [72]. Moreover, model checking suffers from state-explosion problem when analyzing larger systems and some of its underlying algorithms are not formally verified.

Higher-order logic interactive theorem proving [28] is a formal method that provides a conceptually simple formalism with precise semantics. It allows to construct a computer based mathematical model of the system and perform mathematical reasoning to check the systems properties of interest in order to solve the inaccuracy

problem mentioned above. Due to the highly expressive nature of higher-order logic and the inherent soundness of theorem proving, as already described in the previous section, this technique is capable of conducting formal probabilistic analysis, which is the prerequisite of the Markov chain model analysis.

Based on the work of [39] and later [79], in this thesis, we propose a comprehensive framework for the formal analysis of DTMC models by means of higher-order-logic theorem proving. A general overview of the proposed framework is depicted in Figure 1.2.

To conduct the Markovian system analysis, the first step is to construct the formal system model as a function in higher-order logic based on the given system description. This can be done using the left dotted box that contains the formal mathematical definitions of Markov chain foundations including discrete-time Markov chain, classified states, hidden Markov model and the classified discrete-time Markov chain. The second step is to formally express the system properties, which are given as a set of characteristics (system behaviors), as the higher-order logic goals utilizing the formal system model developed in the first step. For the purpose of formally analyzing the system properties (proving these goals), it requires a library containing some pre-verified theorems as the properties based on the general models (shown in the boxes colored as light blue) built upon in the first step. These pre-verified theorems include some classical theorems, such as the *joint probability theorem*, *Chapman-Kolmogorov Equation* and *Absolute probability* [55], etc., and the stationary properties [55], which are verified in higher-order logic based on the classified discrete-time Markov chain. If the system can be described as a hidden Markov model [72], then it requires the verification of related properties in the higher-order-logic theorem prover. Our main work is to establish this library for the purpose of facilitating the formally reasoning

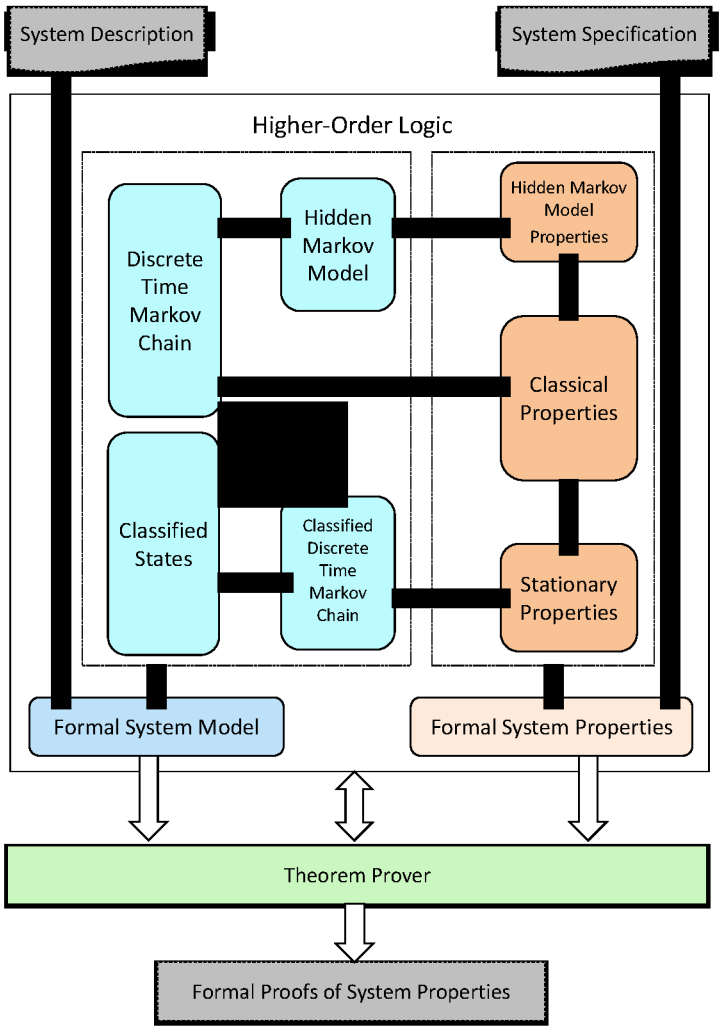

Figure 1.2: Proposed Framework

about Markovian systems. Thus, the third step is to formally verify the goals, developed in the previous steps as a series of theorems, in a theorem prover using the already formally verified theorems. Finally, the output of the theorem prover in this framework is remarked as the formal proofs of system properties and represented by the rectangular box with dashed edges. The output certifies that the given system specification are valid for the given Markovian system.

Based on [39], we developed a formalization of time-homogeneous DTMC with

finite state space [69]. However, the definition of DTMC is not general enough due to the inherit disadvantage of the probability theorem in [49] and the verified theorems are not rich enough to deal with various DTMC models. Our work, in this thesis, mainly relies on the most recent and general formalization of probability theory [81] and all the formalizations of discrete-time Markov chain are executed in theorem prover HOL4 [28].

To the best of our knowledge, the only related work to ours is a very recent work by Hölzl et al. [47], who formally defined a time-homogeneous Markov chain based on the finite state space and the transition matrix in Isabelle/HOL, where the authors they assumed no initial distribution or start state. The aim of their work was to verify PCTL in probabilistic model checkers, hence, a generalized formalization of DTMC theory has not been provided. Furthermore, their work has not shown the capability of formalizing the time-inhomogeneous Markov chain.

1.3 Thesis Contributions

The main contribution of this thesis is to provide an alternative approach to verify Markovian models, which is capable of offering accurate, scalable and generic results. To meet this objective, we construct a foundational framework for conducting Markov chain based analysis within the sound core of the higher-order-logic theorem prover HOL. Some of the key contributions of this thesis are as follows:

- We provide the formalization of DTMC and the verification of the most important properties, in which the concepts of *reversibility* and *stationary properties* accommodate the formal reasoning about *Markov chain mixing time* [67] and the formalizations of *stationary process* and *stationary distribution* grant the formal verification of the stationary properties of numerous other stochastic

processes [Bio-Jr-1, Bio-Cf-1]. We formally analyze a Binary Communication Channel [Bio-Cf-1] and verify certain properties of the Automatic Mail Quality Measurement (AMQM) protocol [Bio-Jr-1]. Also, these formalizations can be built upon to formalize other foundations, such as *Continuous-Time Markov Chain* (CTMC) [Bio-Tr-1].

- We develop the formal definitions of classified states and classified DTMCs, as well as the verified properties of the *aperiodic and irreducible DTMCs*. The properties of aperiodic and irreducible DTMCs can be regarded as theorems and applied in the formal analysis of the long-run behaviors of the Markovian systems [Bio-Cf-2]. These formalizations can be easily used to formally model various interesting stochastic processes and classical problems, such as the *Birth-Death Process* [Bio-Jr-2]. Moreover, these results have been utilized to formally validate a generic Least Recently Used (LRU) stack model [Bio-Cf-2].

- We investigate the formalization of discrete-time HMMs and the verification of their associated properties, such as *joint probability* and the *probability of observation path*, as well as *the best path* obtained in a theorem prover [Bio-Cf-3]. We demonstrate the effectiveness of the foundational theorems through the formal analysis of a DNA sequence in higher-order logic [Bio-Cf-3]. We also stride the first step on proof automation to reduce the human-computer interaction. Thus, providing support to common user, who is not familiar with higher-order logic, to use the theorem prover HOL4 to analyze a HMM, like a DNA sequence.

1.4 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we provide some basic knowledge on probability theory and Markov chain that are required to understand the formalization described in the rest of the thesis. We also offer a brief introduction on theorem proving techniques and the HOL4 theorem prover.

In Chapter 3, we present the proposed higher-order-logic definition of DTMC and use the associated definitions to verify some classical theorems. The formal notations of stationary distribution and stationary process are also presented in this chapter. To illustrate the utilization of these mathematical formalizations, we use them to analyze two applications, namely the analysis of binary communication model and the verification of Automatic Mail Quality Measurement (AMQM) protocol.

In Chapter 4, we present the formalization of diverse classified states and classified DTMCs. Based on their definitions, the major interesting properties of classified DTMCs are formally verified as theorems, which are quite useful in long-term probabilistic analysis. Then, a least recently used (LRU) stack model is formally validated as an application. We also present the formalization of discrete-time Birth-Death chain at the end of this chapter.

As a special Markovian model, the formalization of Hidden Markov Model (HMM) is presented in Chapter 5. The properties, including joint probability and state path probability are verified using the formalization of HMM. We also make use of the proof of best path selection to introduce the automation technique in higher-order logic. In order to show the usefulness of these properties, we present the formal analysis of a DNA sequence.

Finally, Chapter 6 provides concluding remarks and outlines several future research directions.

Chapter 2

Preliminaries

In this chapter, we first briefly introduce higher-order-logic theorem proving and the HOL4 theorem prover. We then present a development of higher-order logic formalization of probability theory in HOL4, which forms the foundational material required to understand this thesis.

2.1 HOL Theorem Proving

Theorem proving is the process of verifying theorems using formal reasoning based on a small set of *axioms* and *inference rules* defined in a proof assistant named *theorem prover*, which provides a sound environment for mathematical reasoning. This sound environment ensures that any new theorem must be verified by applying the basic axioms, primitive inference rules and previously proved theorems. The main idea behind theorem proving is that the system is modeled as a function in some appropriate *logic* and the system properties are expressed as theorems in the same logic, then these theorems are interactively verified based on mathematical reasoning in a theorem prover. The most commonly used logics in the proof systems are *propositional*

logic, *first-order logic* and *higher-order logic*. Among them, propositional logic can be used to automatically reason about complete sentences at the cost of a lower expressiveness; first-order logic can use quantified variables to check individual objects and their relationships, however, the logical consequence relation is semidecidable; whereas higher-order logic requires more interactions because it is neither decidable nor complete, but higher-order logic enables quantification over arbitrary variables, predicates and functions, which provides the capabilities to express any complex systems or reason about any mathematical theory. Hence, higher-order logic has the highest expressiveness. This is the main reason why the theorem proving using higher-order logic is the most flexible verification technique for a wide range of real-world systems. The commonly used theorem provers for higher-order logic are *Coq* [17], *HOL4* [45], *HOL Light* [44], *Isabelle* [50], *MIZAR* [83] and *PVS* [95]. These theorem provers are built upon in sound proof systems, where all theorems are tautologies (all proved theorems are true and semantically valid formulas), while they have the incompleteness feature. However, in order to facilitate the proof process, the higher-order logic theorem prover provides rich proof assistants and automatic proof methods. Due to the undecidable nature of higher-order logic, users have to verify theorems in an interactive way. The most advanced probability theories, on which our work depends, [80] is built upon theorem prover HOL4, hence, we select the theorem prover HOL4 as our proof assistant.

Based on Robin Milner's proof-checking program, *Logic for Computable Functions* (LCF) [82], *HOL4* is developed for conducting proofs in higher-order logic by using the strongly-typed functional Meta-Language (ML) [91]. As a system of deduction with a precise semantics, HOL4 is capable of verifying a wide variety of hardware and software as well as pure mathematics due to the high expressiveness higher-order

logic. One of the key principles of the HOL4 system is that its logical core consists of only 5 axioms and 8 inference rules and all the subsequent theorems are verified based on these foundations or any other previously verified theorems. HOL4 supports both *forward* and *backward* proofs by applying tactics, which are ML functions that simplify goals into subgoals. Over the past few decades, the formalization of many foundational mathematical theories has led to tremendous progress in HOL4. For example, Harrison [36] formalized real numbers, topology, limits, sequences and series as well as differentiation and integration. His work is part of the current distribution of HOL. Hurd [49] developed a probability theory and Hasan [39] formalized statistical properties for continuous random variables and their Cumulative Distribution Function (CDF) in the HOL4 system. However, in Hurd's formalization, the space is implicitly the *universal set* of the appropriate type. Thus, it is not allowed to reason about a measure space where the space is not the universal set. Since Hasan's work is built upon Hurd's, it inherits the same limitations. Later, Coble [15] defined probability spaces and random variables based on an improved measure space, which is formalized as a triple $(X, \mathcal{A}, \mu)$ and contains an arbitrary space to overcome the limitation of Hurd's work. However, in Coble's work, Borel spaces are not defined on open intervals and real-valued measurable functions cannot be defined. More recently, Mhamdi [79] provided a significant formalization of measure theory and probability theory for formally analyzing information theory. It overcomes the limitations of Coble's work by allowing to define sigma-finite and other infinite measures as well as the signed measures. We use this latter formalization for the work in this thesis.

In Table 2.1, we list some frequently used symbols and functions associated with the description in the following chapters of this thesis.

Table 2.1: HOL Symbols and Functions

HOL Symbol	Meaning
$\forall$	Logical *for all*
$\exists$	Logical *exists*
$\wedge$	Logical *and*
$\vee$	Logical *or*
$\sim$	Logical *negation*
(a, b)	A pair of two elements
EL k L	The k^{th} element of list L
REVERSE L	The reverse list of L
divides a b	a can be divided by b
λx.fx	Function that maps x to $f(x)$
{x|P(x)}	Set of all x such that $P(x)$
Univ	Universal Set
$\emptyset$	Empty Set
a $\in$ S	a in S
FINITE S	S is a finite set
A $\subseteq$ B	A is a subset of B
$\bigcap$ P	Intersection of all sets in the set P
$\bigcup$ P	Union of all sets in the set P
A $\cap$ B	A intersection B
A $\cup$ B	A union B
disjoint A B	Sets A and B are disjoint
MAXSET A	The maximum element in a set A
IMAGE f A	Set with elements $f(x)$ for all $x \in A$
convergent (λn. f n)	f is convergent
PROD (0, k) (λn. f n)	$\prod_{n=0}^{k} f(n)$
SIGMA (λn. f n) s	$\sum_{n \in s} f(n)$
suminf (λ n.f n)	$\lim_{k \to \infty} \sum_{n=0}^{k} f(n)$
lim (λ n. f n)	Limit of a *real* sequence f
summable (λ n. f n)	f is summable

2.2 Probability Theory

Mathematically, a *measure space* is defined as a triple (Ω, Σ, μ), where Ω is a set, called the *sample space*, Σ represents a σ-algebra of subsets of Ω, where the subsets are usually referred to as *measurable sets*, and μ is a *measure* with domain Σ. A *probability space* is a measure space $(\Omega, \Sigma, \mathcal{P}r)$ such that the measure, referred to as the probability and denoted by $\mathcal{P}r$, of the sample space is 1. A probability theory is developed based on three axioms [79]:

1. $\forall$ A. $0 \leq \mathcal{P}r(A)$
2. $\mathcal{P}r(\Omega) = 1$
3. For any countable collection A_0, A_1, $\cdots$ of mutually exclusive events, $\mathcal{P}r(\bigcap_{i\in\Omega} A_i) = \sum_{i\in\Omega} \mathcal{P}r(A_i)$.

A *random variable* is a function from a probability space to a *measurable space*. A measurable space refers to a pair (S, Σ), where S denotes a set and Σ represents a nonempty collection of subsets of S. Especially, if the set S is a *discrete set*, which contains only isolated elements, then this random variable is called a *discrete random variable*. The probability that a discrete random variable X is exactly equal to some value i is defined as the *probability mass function* (PMF) and it is mathematically expressed as $\mathcal{P}r(X = i)$.

A *random process* denotes a collection of random variables X_t $(t \in T)$. If the indices (t) of random variables X_t are discrete, then this random process is a *discrete-time random process*.

Nedzusiak [86] and Bialas [9] were among the first to propose to formalize some measure and probability theories in higher-order-logic. Later, Hurd [49] implemented their work and formalized a certain measure space as a pair (Σ, μ) in HOL. The

sample space, on which this pair is defined, is implied from the higher-order-logic definitions to be equal to the universal set of the appropriate data-type. Building upon the formalization of measure space, the probability space was also defined in HOL as a pair $(\mathcal{E}, \mathbb{P})$, where the domain of $\mathbb{P}$ is the set $\mathcal{E}$, which is a set of subsets of infinite Boolean sequences $\mathbb{B}^\infty$. Both $\mathbb{P}$ and $\mathcal{E}$ are defined using the Carathéodory's Extension theorem, which ensures that $\mathcal{E}$ is a σ-algebra: closed under complements and countable unions. As a consequence, the space is implicitly a universal set. Based on Hurd's formalization of probability theory, Hasan [39] provides the formalization of random variables and the verification of statistical properties, such as expectation and variance, for both discrete and continuous random variables [40].

However, as mentioned above, the fact that the sample space in Hurd's measure theory is a universal set results in a very complex definition for the arbitrary space, where the space is not the universal set. This limits its scope considerably.

Later, Coble [16] formalized the measure space as the triple (X, Σ, μ). This allows to define an arbitrary space X, hence overcomeing the disadvantage of Hurd's work. Coble's probability theory is built upon finitely-valued (standard real numbers) measures and functions. Specifically, the Borel sigma algebra cannot be defined on open sets and this constrains the verification of some applications. More recently, Mhamdi [80] improved the development based on the axiomatic definition of probability proposed by Kolmogorov [58]. Mhamdi's theory provides a mathematical consistent for assigning and deducing probabilities of events. Hölzl [46] has also formalized three chapters of measure theory in Isabelle/HOL. Affeldt [2] simplified the formalization of probability theory in Coq [17]. Among these works, the probability theory formalized by Mhamdi provides the most generic formal reasoning support and thus can be used to analyze wider range of applications.

Mhamdi defined a *probability space* in higher-order logic as a measure space $(\Omega, \Sigma, \mathcal{P}r)$ [80], which is exactly matched with the aforementioned mathematical definition . The probability theory is then developed by giving a probability space `p` and the functions `space` and `subsets` which return the corresponding Ω and Σ, respectively. The above approach has been successfully used to formally verify most basic probability theorems [79], such as:

$$0 \leq \mathcal{P}r(B) \leq 1 \tag{2.1}$$

$$\sum_{B_i \in \Omega} \mathcal{P}r(B_i) = 1 \tag{2.2}$$

In [79], a random variable is formally defined (formalized) as a measurable function `X` between a probability space `p` and a measurable space `s`. It is written as `random_variable X p s` in HOL. The definition of random variables is general enough to formalize both discrete and continuous random variables. Now, utilizing the formalization of random variables, the random process $\{X_t\}_{t \geq 0}$ can be easily written as $\forall$ `t. random_variable (X t) p s` in higher-order logic.

One of the crucial concepts in the random process study is the *conditional probability*, which is used to calculate the occurrence probability of an event when another event is known to occur. Conditional probability basically reflects the dependency between the events which happen at different times in a process. The formal definition of conditional probability in HOL can be found in [41], which is based on Hurd's work [49]. In order to make use of the most advanced probability theory in our work, we improved the formalization of conditional probability as:

Definition 2.1. *(Conditional Probability)*

The conditional probability of the event A given the occurrence of the event B is

$$\mathcal{P}r(A|B) = \mathcal{P}r(A \cap B)/\mathcal{P}r(B)$$

$\vdash$ $\forall$ `A B. cond_prob p A B = prob p (A ∩ B) / prob p B`

where `cond_prob` represents conditional probability, and `prob` denotes the probability. There are different functions of a probability space p in HOL. In this thesis, we utilize the symbol $\mathbb{P}$ to denote both the function `cond_prob p` and the function `prob p` in HOL and the argument of $\mathbb{P}$ would clarify if we want to use it in the context of `cond_prob` (e.g. $\mathbb{P}$(A | B)) or `prob` (e.g. $\mathbb{P}$(B)).

In order to facilitate the formalization of Markov chains, we verified various classical properties of conditional probability based on Definition 2.1. Some of the prominent ones are listed below:

$$\mathcal{P}r(A \cap B) = \mathcal{P}r(A|B)\mathcal{P}r(B) \tag{2.3a}$$

$$\mathcal{P}r(A \cap B|C) = \mathcal{P}r(A|B \cap C)\mathcal{P}r(B|C) \tag{2.3b}$$

$$\mathcal{P}r(A) = \textstyle\sum_{i\in\Omega} \mathcal{P}r(B_i)\mathcal{P}(A|B_i) \tag{2.3c}$$

$$\mathcal{P}r(A) = \textstyle\sum_{i\in\Omega} \mathcal{P}r(A)\mathcal{P}r(B_i|A) \tag{2.3d}$$

$$\textstyle\sum_{i\in\Omega} \mathcal{P}r(B_i|A) = 1 \tag{2.3e}$$

where A, B and C are events in the event space, and the finite events set $\{B_i\}_{i\in\Omega}$ contains mutually exclusive and exhaustive events. The first two theorems are obviously based on Definition 2.1. The third one is the Total Probability Theorem and the fourth one is a lemma of the Total Probability Theorem. The last theorem is the Additivity Theorem.

Mathematically, the *conditional independence* [53] is an important concept, which is the foundation of graphical models and mainly used in Bayesian Network. The mathematical definition of conditional independence is:

Definition 2.2. *(Conditional Independence)*

The events A and B are conditionally independent given the event C if

$$\mathcal{P}r(A|B \cap C) = \mathcal{P}r(A|C). \tag{2.4}$$

Chapter 3

Discrete-time Markov Chain in HOL

In this chapter, we describe the formalization of discrete-time Markov chain and the formal verification of some of its most important properties using the probability theory presented in [80]. In order to illustrate the usefulness of this work, a binary communication channel model and a collection process model involved in the Automatic Mail Quality Measurement (AMQM) protocol are formally analyzed in HOL.

3.1 Definition of Discrete-time Markov Chain

Given a probability space, a stochastic process $\{X_t : \Omega \to S\}$ represents a sequence of random variables X, where t represents the time that can be discrete (represented by non-negative integers) or continuous (represented by real numbers) [8]. The set of values taken by each X_t, commonly called states, is referred to as the *state space*. The *sample space* Ω of the process consists of all the possible state sequences based on a given state space S. Now, based on these definitions, a *Markov process* can be

defined as a stochastic process with *Markov property* [13]. If a Markov process has finite or countably infinite state space Ω, then it is called a *Markov chain* and satisfies the following Markov property:

For $0 \leq t_0 \leq \cdots \leq t_n$ and $f_0, \cdots, f_{n+1}$ in the state space, then:

$$\mathcal{P}r\{X_{t_{n+1}} = f_{n+1} | X_{t_n} = f_n, \ldots, X_{t_0} = f_0\} = \mathcal{P}r\{X_{t_{n+1}} = f_{n+1} | X_{t_n} = f_n\} \qquad (3.1)$$

This means that the future state is only dependent on the current state and is independent of all the other past states. Now, the Markov property can be formalized as follows:

Definition 3.1. *(Markov Property)*

```
⊢ ∀ X p s.  mc_property X p s =
  (∀ t.  random_variable (X t) p s) ∧
  ∀ f t n.
   increasing_seq t ∧ ℙ(⋂_{k∈ [0,n−1]}{x | X t_k x = f k}) ≠ 0 ⇒
   (ℙ({x | X t_{n+1} x = f (n + 1)}|{x | X t_n x = f n} ∩
                                    ⋂_{k∈ [0,n−1]}{x | X t_k x = f k}) =
    ℙ({x | X t_{n+1} x = f (n + 1)}|{x | X t_n x = f n}))
```

where `increasing_seq t` is defined as `∀ i j.  i < j ⇒ t i < t j`, thus formalizing the notion of increasing sequence. The first conjunct indicates that the Markov property is based on a random process $\{X_t : \Omega \to S\}$. The quantified variable `X` represents a function of the random variables associated with time `t` which has the type `num`. This ensures the process is a *discrete time* random process. The random variables in this process are the functions built on the probability space `p` and a measurable space `s`. The conjunct $\mathbb{P}(\bigcap_{k\in [0,n-1]}\{x \mid X\ t_k\ x = f\ k\}) \neq 0$ ensures that

the corresponding conditional probabilities are well-defined, where `f k` returns the $\texttt{k}^{th}$ element of the state sequence.

We also have to explicitly mention all the usually implicit assumptions stating that the states belong to the considered space. The assumption $\mathbb{P}(\bigcap_{\texttt{k}\in\texttt{ts}}\{\texttt{x | X k x = f k}\}) \neq \texttt{0}$ ensures that the corresponding conditional probabilities are well-defined, where `f k` returns the $\texttt{k}^{th}$ element of the state sequence. In fact, the assumption $\mathbb{P}(\bigcap_{\texttt{k}\in[\texttt{0},\texttt{n}-\texttt{1}]}\{\texttt{x | X t}_{\texttt{k}}\texttt{ x = f k}\}) \neq \texttt{0}$ ensures that the corresponding conditional probabilities are well-defined and represents the following HOL code:

```
prob p (BIGINTER (IMAGE (λ k.  (X t_k x = f k) ∧ (x IN p_space p)) [0, n - 1]))
```

The term `x IN p_space p` ensures that `x` is in the samples space in the considered probability space `p_space p`.

For better clarity, in this thesis, $\mathbb{P}\{\texttt{x | X t x = i}\}$ represents `prob p {x | (X t x = i) ∧ (x IN p_space p)}`. Similarly, the conditional probability $\mathbb{P}(\{\texttt{x | X (t + 1) x = j}\}|\{\texttt{x | X t x = i}\})$ represents `cond_prob p ({x | (X (t + 1) x = j) ∧ (x IN p_space p)}|{x | (X t x = i) ∧ (x IN p_space p)})` in HOL.

A *DTMC with finite state space* is usually expressed by specifying: an initial distribution p_0 which gives the probability of initial occurrence $\mathcal{P}r(X_0 = s) = p_0(s)$ for every state; and transition probabilities $p_{ij}(t)$ which give the probability of going from i to j for every pair of states i, j in the state space [88]. For states i, j and a time t, the *transition probability* $p_{ij}(t)$ is defined as $\mathcal{P}r\{X_{t+1} = j|X_t = i\}$, which can be easily generalized to *n-step transition probability.*

$$p_{ij}^{(n)}(t) = \begin{cases} \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} & n = 0 \\ \mathcal{P}r\{X_{t+n} = j|X_t = i\} & n > 0 \end{cases} \tag{3.2}$$

This is formalized in HOL as follows:

Definition 3.2. *(Transition Probability)*

```
⊢ ∀ X p s t n i j.  Trans X p s t n i j =
  if i ∈ space s ∧ j ∈ space s then
    if n = 0 then
      if (i = j) then 1
      else 0
    else ℙ({x | X (t + n) x = j}|{x | X t x = i})
  else 0
```

It is easy to understand that the probability of an event is zero, when this event is not in the event space. For instance, i is not in the state space implies that event $\{X_t = i\} = \emptyset$. In this case, the conditional probability related to an empty set is zero.

Now, the discrete-time Markov chain (DTMC) can be formalized as follows:

Definition 3.3. *(Discrete-Time Markov Chain)*

```
⊢ ∀ X p s p₀ pᵢⱼ.
  dtmc X p s p₀ =
  mc_property X p s ∧ (∀ i.  i ∈ space s ⇒ {i} ∈ subsets s) ∧
  ∀ i.  i ∈ space s ⇒ (p₀ i = ℙ{x | X t x = i}) ∧
  ∀ t i j.  ℙ{x | X t x = i} ≠ 0 ⇒ (pᵢⱼ t i j = Trans X p s t 1 i j)
```

where the first three variables are inherited from Definition 3.1, p_0 and p_{ij} refer to the functions expressing the given initial status and transition matrix associated with this random process, respectively. The first condition in this definition describes Markov property presented in Definition 3.1 and the second one ensures the events associated

with the state space (`space s`) are discrete in the event space (`subsets s`), which is a *discrete space*. The last two conditions assign the functions $\mathtt{p_0}$ and $\mathtt{p_{ij}}$ to initial distributions and transition probabilities.

It is important to note that `X` is polymorphic, i.e., it is not constrained to a particular type, which is a very useful feature of our definition.

In Definition 3.3, if the function p_{ij} depends on `t`, then this discrete-time Markov chain is a time-inhomogeneous Markov chain. However, most of the applications actually make use of *time-homogenous DTMCs*, i.e., DTMCs with finite state-space and time-independent transition probabilities [5]. The time-homogenous property refers to the time invariant feature of a random process. Thus, the one-step transition probability of the random process can be simplified as $p_{ij} = \mathcal{P}r\{X_{t+1} = j | X_t = i\} = p_{ij}(t)$, based on Equation (3.2).

Then, the time-homogenous DTMC with finite state-space can be formalized as follows:

Definition 3.4. *(Time homogeneous DTMC)*

```
⊢ ∀ X p s p₀ pᵢⱼ.
    th_dtmc X p s p₀ pᵢⱼ =
    dtmc X p s p₀ pᵢⱼ ∧ FINITE (space s) ∧
    ∀ t i j.  ℙ{x | X t x = i} ≠ 0 ∧ ℙ{x | X (t + 1) x = i} ≠ 0 ⇒
      (Trans X p s (t + 1) 1 i j = Trans X p s t 1 i j)
```

where the first and second conjuncts constraint this time-homogeneous DTMC is a discrete-time Markov chain with the finite state space, the last condition expresses the time-homogeneous property: $\forall\ t\ t'.\ p_{ij}(t) = p_{ij}(t')$ and thus $p_{ij}(t)$ is simply written as p_{ij} in the rest of this thesis.

3.2 Classical DTMC Properties

Using the formal definition of DTMC, we proved the most important properties of DTMC, which are usually called *classical* DTMC properties and are frequently used in the analysis of many systems modeled as DTMCs.

3.2.1 Joint Probability Theorem

The *joint probability distribution* of a DTMC is the probability of a chain of states to occur. It is very useful in analyzing multi-stage experiments, especially, the stationary process which is defined on the joint probability distribution. The joint probability distribution is also used to analyze the reversibility of a Markov chain. Moreover, this concept is the basis for the joint probability generating function, which is used in many different fields. We verified this property in HOL as the following theorem:

Theorem 3.1. *(Joint Probability)*

A joint probability distribution of n discrete random variables $X_0, \ldots X_n$ *in a finite DTMC* $\{X_t\}_{t\geq 0}$ *satisfies:*

$$\mathcal{P}r(X_t = L_0, \cdots, X_{t+n} = L_n) = \prod_{k=0}^{n-1} (\mathcal{P}r(X_{t+k+1} = L_{k+1} | X_{t+k} = L_k)) \mathcal{P}r(X_t = L_0)$$

```
⊢ ∀ X p s t L p₀ pᵢⱼ.
   dtmc X p s p₀ pᵢⱼ ⇒
   ℙ(⋂ⁿₖ₌₀{x | X (t + k) x = EL k L}) =
   (PROD (0, n) (λ k.  ℙ({x | X (t + k + 1) x = EL (k + 1) L}|
                          {x | X (t + k) x = EL k L})))
   ℙ{x | X t x = EL 0 L}
```

Proof. We proved Theorem 3.1 by performing induction on the variable `n`. The base case can be easily verified by using some set and real arithmetic reasoning. The proof of the step case starts from rewriting the $\mathbb{P}(\bigcap_{k=0}^{n+1}\{$`x | X (t + k) x = EL k L`$\})$ to be $\mathbb{P}(\bigcap_{k=0}^{n}\{$`x | X (t + k) x = EL k L`$\} \cap \{$`x | X (t + n + 1) x = EL (n + 1) L`$\})$. Then, the proof can be completed by applying Equation (2.3a) and rewriting the subgoal with Definition 3.3 and the assumption obtained after proving the base case.

3.2.2 Chapman-Kolmogorov Equation

The *Chapman-Kolmogorov Equation* [8] is a widely used property of time homogeneous DTMCs. It gives the probability of going from state i to j in $m + n$ steps. Assuming the first m steps take the system from state i to some intermediate state k and the remaining n steps then take the system from state k to j, we can obtain the desired probability by adding the probabilities associated with all the intermediate steps.

Theorem 3.2. *(Chapman-Kolmogorov Equation)*

For a finite time homogeneous DTMC $\{X_t\}_{t\geq 0}$*, its transition probabilities satisfy the Chapman-Kolmogorov Equation*

$$p_{ij}^{(m+n)} = \sum_{k\in\Omega} p_{ik}^{(m)} p_{kj}^{(n)}$$

```
⊢ ∀ X p s i j t m n p₀ pᵢⱼ.
  th_dtmc X p s p₀ pᵢⱼ ⇒
  (Trans X p s t (m + n) i j =
   SIGMA (λ k.  Trans X p s (t + n) m i k * Trans X p s t n k j)
         (k ∈ space s))
```

Proof. Theorem 3.2 is again verified using induction on the variables m and n. Both of the base and step cases are discharged using the following lemma:

Lemma 3.1. *(Multistep Transition Probability)*

```
⊢ ∀ X p s i j t m p₀ pᵢⱼ.
  th_dtmc X p s p₀ pᵢⱼ ⇒
  (Trans X p s t (m + 1) i j =
   SIGMA (λ k.  Trans X p s (t + m) 1 k j * Trans X p s t m i k)
         (k ∈ space s))
```

which gives the m step transition probability $p_{ij}^{(m)} = \sum_{k\in\Omega} p_{ik}^{(m)} p_{kj}$. The proof of Lemma 3.1 starts from rewriting the goal using Definitions 3.2 and 3.4, and then simplifying the subgoal using the additivity property of probabilities.

3.2.3 Absolute Probability

The unconditional probabilities associated with a Markov chain are called *absolute probabilities*, which can be computed by applying the initial distributions and n-step transition probabilities. This shows that, given the initial probability distributions and the n-step transition probabilities, the absolute probability in the state j after n steps from the start time 0 can be obtained by using this equation.

This property is formally verified as the following theorem:

Theorem 3.3. *(Absolute Probability)*

In a finite time homogeneous DTMC, the absolute probabilities $p_j^{(n)}$ *satisfy*

$$p_j^{(n)} = \mathcal{P}r(X_n = j) = \sum_{k\in\Omega} \mathcal{P}r(X_0 = k)\mathcal{P}r(X_n = j | X_0 = k)$$

```
⊢ ∀ X p s j n p₀ pᵢⱼ.
  th_dtmc X p s p₀ pᵢⱼ ⇒
```

```
(ℙ{x | X n x = j} =
 SIGMA (λ k.  ℙ{x | X 0 x = k}
              ℙ({x | X n x = j}|{x | X 0 x = k})) (k ∈ space s)))
```

The proof of Theorem 3.3 is based on the Total Probability theorem (2.3c) along with some basic arithmetic and probability theoretic reasoning.

3.2.4 Reversibility Property

A stochastic process $\{X_t\}_{t\geq 0}$ is said to be reversible if the joint probability of the sequence X_0, X_1, ..., X_n is the same as the distribution of X_n, X_{n-1}, ..., X_0, that is:

$$\mathcal{P}r\{X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t\} = \mathcal{P}r\{X_t = x_0, X_{t-1} = x_1, \ldots, X_0 = x_t\}.$$

This reversible stochastic process can be defined in HOL as follows:

Definition 3.5. *(A Reversible Stochastic Process)*

```
⊢ ∀ X p s L t.

  reversible_proc X p s =
  (ℙ(⋂_{k=0}^{|L|−1} {X_{t+k} = EL k L}) = ℙ(⋂_{k=0}^{|L|−1} {X_{t+k} = EL k (REVERSE L)}))
```

where `|L|` represents the length of the chain considered.

In Markov chain theory, certain Markov chains satisfy the *Detailed Balance Equations* [88], which ensures that this Markov chain has an equilibrium distribution. This distribution is usually utilized in MCMC methods to construct the samples with a desired distribution in excessive simulations. This kind of Markov chain also exhibits a reversibility feature and is called *reversible Markov chain*. Mathematically, the Detailed Balance Equations are expressed as

$$\pi(i)P_{ij} = \pi(j)P_{ji}, \forall i, j \in \Omega \tag{3.3}$$

where the $\pi(i)$ and $\pi(j)$ are the equilibrium probabilities of being in states i and j, respectively. The detailed balance equations can be formalized as:

Definition 3.6. *(Detailed Balance Equations)*

```
⊢ ∀ f X p s.  db_equations f X p s =
  ∀ i j t.  (f i * Trans X p s t 1 i j = f j * Trans X p s t 1 j i)
```

The function `f` represents the probability distribution π in Equation (3.3). Note that if i or j is not in the state space, then it can imply both the transition probabilities P_{ij} and P_{ji}, corresponding to `Trans X p s t 1 i j` and `Trans X p s t 1 j i`, respectively, equal to zero and thus both sides of Equation (3.3) are zeroes.

A Markov chain is defined as *reversible Markov chain* if it satisfies the Detailed Balance Equations. The formal definition in HOL is

Definition 3.7. *(Reversible DTMC)*

```
⊢ ∀ X p s p₀ pᵢⱼ.
  rmc X p s p₀ pᵢⱼ =
  th_dtmc X p s p₀ pᵢⱼ ∧ ∀ t.  db_equations (λ i.  ℙ{X_t = i}) X p s
```

The concept of the reversible process is mainly applied in the area of thermodynamics [65], while reversible Markov chains are commonly used in MCMC approaches [25]. We can prove the following property of the reversible Markov chain.

Theorem 3.4. *(Reversibility Property)*

```
⊢ ∀ X p s p₀ pᵢⱼ t L.
  rmc X p s p₀ pᵢⱼ ⇒
  (ℙ(⋂_{k=0}^{|L|−1} {X_{t+k} = EL k L}) = ℙ(⋂_{k=0}^{|L|−1} {X_{t+k} = EL k (REVERSE L)}))
```

Proof. The backward proof of this theorem starts from rewriting the goal by using Definition 3.7 and Theorem 3.1, we obtain the following equation in the subgoal to be proved:

```
PROD (0, |L|)
     (λk.  ℙ({x | X (t + k + 1) x = EL (k + 1) (REVERSE L)}|
             {x | X (t + k) x = EL k (REVERSE L)})) =
PROD (0, |L|)
     (λk.  ℙ({x | X (t + k + 1) x = EL (|L| - k - 1) L}|
             {x | X (t + k) x = EL (|L| - k) L}))
```

Then the proof can be completed by applying the following lemma:

```
⊢ ∀ f n.  PROD (0, n) (λ k.  f k) = PROD (0, n) (λ k.  f (n - k - 1))
```

which is corresponding to the mathematical expression:

$$\prod_{k=0}^{n} f(k) = \prod_{k=0}^{n} f(n-k-1)$$

3.3 Stationary Distributions

It is often the case that we are interested in the probability of some specific states as time tends to infinity under certain conditions. This is the main reason why stationary behaviors of stochastic processes are frequently analyzed in engineering and scientific domains. There is no exception for DTMCs.

Let $\{\mathrm{X}_t\}_{t\geq 0}$ be a Markov chain having state space Ω and transition probabilities $\{p_{ij}\}_{i,j\in\Omega}$. If $\pi(i)$, $i \in \Omega$, are nonnegative numbers summing to one, then $\pi(j) = \sum_{i\in\Omega} \pi(i)p_{ij}$ is called a *stationary distribution*. The corresponding HOL definition is as follows.

Definition 3.8. *(Stationary Distribution)*

```
⊢ ∀ f X p s.
   stationary_dist f X p s =
   (SIGMA (λ k.  f k) (k ∈ space s) = 1) ∧
   ∀ i.  i ∈ space s ⇒
         0 ≤ f i ∧
         (∀ t.  f i = SIGMA (λ k.  f k * Trans X p s t 1 k i)
                            (k ∈ space s))
```

We then utilize this definition to prove the *generalized stationary theorem* in HOL:

Theorem 3.5. *(Generalized Stationary Distribution)*

If a DTMC with finite state space Ω and one-step transition probability P_{ij} has a probability distribution π that satisfied the detailed balance equations, then there exits a stationary distribution for this DTMC.

```
⊢ ∀ X p s p₀ pᵢⱼ n.
   th_dtmc X p s p₀ pᵢⱼ ∧ db_equations (λ i.ℙ{x | X n x = i}) X p s ⇒
   ∃ f.  stationary_dist f X p s
```

where `f` refers to $\pi(x)$.

Proof. The proof can be completed by specifying `f` as a function (`λ i.  ℙ{x | X n x = i}`), which is a probability distribution. Then the goal becomes to

```
⊢ ∀ X p s p₀ pᵢⱼ n.
   th_dtmc X p s p₀ pᵢⱼ ∧ db_equations (λ i.ℙ{x | X n x = i}) X p s ⇒
   stationary_dist (λ i.ℙ{x | X n x = i}) X p s
```

Proof. We start to prove this subgoal by rewriting the detailed balance equations and stationary distribution using Definition 3.6 and 3.8. Then the goal is split into three subgoals.

- `0 ≤ ℙ{x | X n x = i}`
- `SIGMA (λ j. ℙ{x | X n x = i} (j ∈ space s) = 1)`
- `ℙ{x | X n x = i} =`
  `SIGMA (λ j. ℙ{x | X n x = j} Trans X p s t j i) (j ∈ space s)`

The first subgoal can be proved by applying the probability theorem in expression (2.1). While the second one is completed by using the probability additivity theorem shown in Equation (2.2). The last subgoal is proved by rewriting it using the Equation (2.3d) and Definition 3.6.

In a time-homogeneous DTMC with finite state space Ω and one-step transition probability $\{p_{ij}\}_{i,j\in\Omega}$, the steady state probabilities are defined in a vector V_j, which equals to $\lim\limits_{n\to\infty} p_j(n)$, if the limiting probability exists for any $j \in \Omega$. If a system is in a steady state, numerous properties are of interest due to the reason that they are invariant with time.

If the limiting probability exists for any $j \in \Omega$ in the state space of such a time-homogeneous DTMC, then the steady state probability vector V_j is the stationary probability vector of that Markov chain. In other words, V_j is a stationary distribution of this DTMC if it satisfies:

- $V_j = \sum\limits_{i \in \Omega} \lim_{n\to\infty} V_j p_{ij}$
- $\sum_{i \in \Omega} V_i = 1$
- $0 \leq \lim_{n\to\infty} V_j$

These three conditions exactly correspond to the subgoals listed in the proof of Theorem 3.5. This stationary property can be verified using higher-order logic as the following theorem:

Theorem 3.6. *(Steady State Probabilities)*

```
⊢ ∀ X p s p₀ pᵢⱼ n.
  th_dtmc X p s p₀ pᵢⱼ ∧
  db_equations (λ i.  lim ℙ{x | X n x = i}) X p s ⇒
  stationary_dist (λ i.  lim ℙ{x | X n x = i}) X p s
```

Proof. After rewriting the goal with Definition 3.8, the proof process is similar as that of Theorem 3.5.

3.4 Stationary Process

Stationary processes are frequently used stochastic processes for analyzing time series, which are the foundations of Ergodic theorems [55]. Mathematically, a stochastic process $\{X_t\}_{t\in T}$ is said to be stationary in the strict sense, if for $n \geq 1$, t_1, t_2, ..., t_n, $\tau \in$ T, the random variables X_{t_1}, X_{t_2}, ..., X_{t_n} have the same joint distributions as $X_{t_1+\tau}$, $X_{t_2+\tau}$, ..., $X_{t_n+\tau}$. In a discrete-time stochastic process, τ is a natural number. It is worth to note that a stationary process is different from a process with stationary distribution. In HOL, we formalize a stationary process as follows:

Definition 3.9. *(Stationary Process)*

```
⊢ ∀ X p s.
  stationary_proc X p s =
  ∀ f t w n.  ∀ t.  random_variable (X t) p s ∧
```

$$(\mathbb{P}(\bigcap_{k=0}^{n}\{\texttt{x} \mid \texttt{X (w + k) x = f k}\}) =$$
$$\mathbb{P}(\bigcap_{k=0}^{n}\{\texttt{x} \mid \texttt{X (t + k) x = f k}\}))$$

where `f` denotes the state sequence.

Using this definition, we can apply induction on variables `t` and `n` to prove that the PMF of a stationary process is independent of the time.

Theorem 3.7. *(Stationary Process Property)*

$\vdash \forall$ `X p s t n j.`

`stationary_pmf X p s` $\Rightarrow$ $(\mathbb{P}(\{$`x | X n x = j`$\}) = \mathbb{P}(\{$`x | X t x = j`$\}))$

A stationary process is not a process with a stationary distribution. In fact, a DTMC is stationary if and only if its initial distribution is stationary. We formally verify this property from two different perspectives, as shown in the following two theorems.

Theorem 3.8. *(Stationary DTMC has Stationary Distribution)*

A stationary DTMC has stationary distributions for all the states.

$\vdash \forall$ `X p s n` $\texttt{p}_0$ $\texttt{p}_{\texttt{ij}}$.

`th_dtmc X p s` $\texttt{p}_0$ $\texttt{p}_{\texttt{ij}}$ $\wedge$ `stationary_pmf X p s` $\Rightarrow$

`stationary_dist` $(\lambda$ `i.`$\mathbb{P}\{$`x | X n x = i`$\})$ `X p s`

Proof. We rewrite the goal using Definitions 3.8 and 3.9 and then split the goal into three subgoals and the proof steps become similar to those for Theorem 3.5. The proof of the last subgoal requires $\mathbb{P}\{$`x | X n x = j`$\} = \mathbb{P}\{$`x | X (n - 1) x = j`$\}$, which can be proved by instantiating the variables `t` and `n` to be `n` and `n - 1`, respectively, using Theorem 3.7. The proof is finalized by applying Theorem 3.3.

If the variable n in Theorem 3.8 is assigned a value 0 then the stationary DTMC is said to have a *stationary initial distribution*. Thus, Theorem 3.8 shows that a stationary DTMC has stationary initial distributions.

Theorem 3.9. *(A DTMC with Stationary Initial Distribution is a Stationary Process)*

If the initial distributions of a DTMC are stationary then the corresponding DTMC is stationary as well.

$$
\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ s\ p_0\ p_{ij}}.\\
&\quad \mathtt{th\_dtmc\ X\ p\ s\ p_0\ p_{ij}} \wedge\\
&\quad \mathtt{stationary\_dist\ (\lambda\ i.\mathbb{P}\{x\ |\ X\ 0\ x = i\})\ X\ p\ s} \Rightarrow\\
&\quad \mathtt{stationary\_proc\ X\ p\ s}
\end{aligned}
$$

Proof. We proceed with the verification of this theorem by first rewriting the goal using Definitions 3.8 and 3.9 and then performing induction on the variable n of the stationary process definition, given in Definition 3.9. The base case is true obviously and the step case can be proved using Theorem 3.1.

Using these fundamental definitions, we formally verified most of the classical properties of DTMCs with finite state-space in HOL. Some of the relevant ones to the context of this thesis are presented later. In next section, we provide two applications of the above definitions and theorems.

3.5 Applications

To illustrate the usefulness of the formalization of DTMC in higher-order logic, in this section we present two applications: a simplified binary communication channel [109] and the Automatic Mail Quality Measurement (AMQM) protocol [87].

3.5.1 Binary Communication Model

A binary communication channel [109] is a channel with binary inputs and outputs. The transmission channel is assumed to be noisy or imperfect, i.e., it is likely that the receiver gets the wrong digit. This channel can be modeled as a two-state DTMC with the following state transition probabilities.

$\mathcal{P}\mathtt{r}\{X_{n+1} = 0 \mid X_n = 0\} = 1 - a;$ $\quad$ $\mathcal{P}\mathtt{r}\{X_{n+1} = 1 \mid X_n = 0\} = a;$

$\mathcal{P}\mathtt{r}\{X_{n+1} = 0 \mid X_n = 1\} = b;$ $\quad$ $\mathcal{P}\mathtt{r}\{X_{n+1} = 1 \mid X_n = 1\} = 1 - b$

The corresponding state and channel diagrams are given in Figure 3.1 and 3.2, respectively.

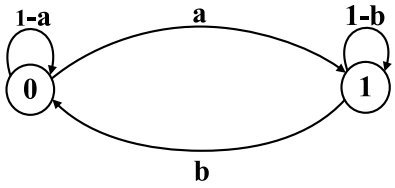

Figure 3.1: State Diagram

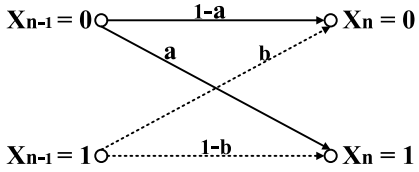

Figure 3.2: Channel Diagram

The binary communication channel is widely used in telecommunication theory as more complicated channels are modeled by cascading several of such channel. In Figure 3.2, variables X_{n-1} and X_n denote the digits leaving the systems $(n-1)^{th}$ stage and entering the n^{th} one, respectively. a and b are the crossover bit error probabilities. Because X_0 is also a random variable, the initial state cannot be determined and thus the initial distributions $\mathcal{P}\mathrm{r}(X_0 = 0)$ and $\mathcal{P}\mathrm{r}(X_0 = 1)$ cannot be 0 or 1. Although the initial distribution is unknown, the n-step transition probabilities can be verified as the elements of the matrix in Equation (3.4). Also, the steady-state probabilities can be concluded as that in Equation (3.5).

$$P^n = \begin{pmatrix} \frac{b+a(1-a-b)^n}{a+b} & \frac{a-a(1-a-b)^n}{a+b} \\ \frac{b-b(1-a-b)^n}{a+b} & \frac{a+b(1-a-b)^n}{a+b} \end{pmatrix} \tag{3.4}$$

$$\lim_{n\to\infty} P^n = \begin{pmatrix} \frac{b}{a+b} & \frac{a}{a+b} \\ \frac{b}{a+b} & \frac{a}{a+b} \end{pmatrix} \tag{3.5}$$

In HOL, we defined two functions `Linit` and `Lt` to express the initial distributions and transition probabilities.

Definition 3.10. *(Initial Distributions)*

```
⊢ ∀ c d.
   Linit c d i =
     if (i = 0) then c
     else
       if (i = 1) then d
       else 0
```

Definition 3.11. *(Transition Probabilities)*

```
⊢ ∀ a b t i j.
   Lt a b t i j =
     if (i = 0) ∧ (j = 0) then 1 - a
     else
       if (i = 0) ∧ (j = 1) then a
       else
         if (i = 1) ∧ (j = 0) then b
         else
           if (i = 1) ∧ (j = 1) then 1 - b
           else 0
```

Based on the description of the binary communication channel, it is formalized in HOL, using Definition 3.12.

Definition 3.12. *(Binary Communication Channel Model)*

```
⊢ ∀ X p a b c d.
   BINARY_CHANNELS_MODEL X p a b c d =
   th_dtmc X p ([0, 1], POW [0,1]) (Linit c d) (Lt a b) ∧
   |1 - a - b| < 1 ∧ 0 ≤ a ∧ a ≤ 1 ∧ 0 ≤ b ∧ b ≤ 1 ∧
    (c + d = 1) ∧ 0 < c ∧ c < 1 ∧ 0 < d ∧ d < 1
```

In this formal model, the function `X` represents the random variable in the time-homogeneous DTMC. `p` represents the probability space and the measurable state space is expressed as a pair `([0, 1], POW [0,1])`, where "`0`" and "`1`" are involved in the set of state value, and the second element `POW [0,1]` is a sigma-algebra. Variables `a`, `b`, `c` and `d` are parameters of the functions of the initial distributions and transition probabilities.

The first condition ensures that the channel can be modeled as a time-homogeneous DTMC, with two states in the state space. `Linit c d` and `Lt a b` represent the general initial distributions and the transition probabilities (corresponding to the p_0 and p_{ij} in Definition 3.3), respectively. The next five conditions define the allowable intervals for parameters `a` and `b` to restrict the probability terms in `[0,1]`. It is important to note that, `|1 - a - b| < 1` ensures that both `a` and `b` cannot be equal to 0 and 1 at the same time and thus avoids the zero transition probabilities. The remaining conditions correspond to the boundary of parameters `c` and `d`, which are probabilities and would not be determined as "0" or "1".

Next, we use our formal model to reason about the following properties, which correspond to Equations (3.4) and (3.5).

Theorem 3.10. *(n^{th} step Transition Probabilities)*

```
⊢ ∀ X p a b c d n.
  BINARY_CHANNELS_MODEL X p a b c d ⇒
```

$$(\mathbb{P}(\{x \mid X\ n\ x = 0\} \mid \{x \mid X\ 0\ x = 0\}) = \tfrac{b+a(1-a-b)^n}{a+b}) \wedge$$
$$(\mathbb{P}(\{x \mid X\ n\ x = 1\} \mid \{x \mid X\ 0\ x = 0\}) = \tfrac{a-a(1-a-b)^n}{a+b}) \wedge$$
$$(\mathbb{P}(\{x \mid X\ n\ x = 0\} \mid \{x \mid X\ 0\ x = 1\}) = \tfrac{b-b(1-a-b)^n}{a+b}) \wedge$$
$$(\mathbb{P}(\{x \mid X\ n\ x = 1\} \mid \{x \mid X\ 0\ x = 1\}) = \tfrac{a+b(1-a-b)^n}{a+b})$$

Proof. Theorem 3.10 has been verified by rewriting the original goal using Theorem 3.2 and performing induction on `n` and then completing the proof by applying some arithmetic reasoning.

Theorem 3.11. *(Limiting State Probabilities)*

$$\vdash \forall\ X\ p\ a\ b\ c\ d.$$
$$\texttt{BINARY\_CHANNELS\_MODEL}\ X\ p\ a\ b\ c\ d \Rightarrow$$
$$(\lim\ (\lambda\ n.\ \ \mathbb{P}(\{x \mid X\ n\ x = 0\} \mid \{x \mid X\ 0\ x = 0\}) = \tfrac{b}{a+b}) \wedge$$
$$(\lim\ (\lambda\ n.\ \ \mathbb{P}(\{x \mid X\ n\ x = 1\} \mid \{x \mid X\ 0\ x = 0\}) = \tfrac{a}{a+b}) \wedge$$
$$(\lim\ (\lambda\ n.\ \ \mathbb{P}(\{x \mid X\ n\ x = 0\} \mid \{x \mid X\ 0\ x = 1\}) = \tfrac{b}{a+b}) \wedge$$
$$(\lim\ (\lambda\ n.\ \ \mathbb{P}(\{x \mid X\ n\ x = 1\} \mid \{x \mid X\ 0\ x = 1\}) = \tfrac{a}{a+b})$$

Proof. Theorem 3.10 is then used to verify Theorem 3.11 along with the limit of real sequence principles.

A special case of the property corresponding to Equation (3.4) has been verified in the PRISM model checker for some specific values of the variables `a`, `b`, `c`, `d` and `n` [71]. However, Equation (3.5) cannot be verified directly in a PRISM due to the involvement of limiting behavior. Unlike model checking, our approach provides the verification of the desired probabilistic characteristics as generic theorems (i.e., Theorems 3.10 and 3.11) that are universally quantified for all allowable values of variables. This small two-state DTMC case study clearly illustrates the main strength of the proposed theorem proving based technique against the probabilistic model checking [94] approach.

3.5.2 AMQM Protocol

In this section, we will study the probability of reaching a targeted state in an Automatic Mail Quality Measurement (AMQM) system based on the ISO/IEC 18000-7 Standard [1] by building upon our formalized DTMC described above.

An AMQM system is used to measure the quality of postal service transport and delivery by IPC (International Post Corporation). It measures how fast mail travels from one point to another by using an in-planting process monitoring of the tag serial number and recording the time when a message from the tag is received. This kind of quality measurement of solutions is based on Radio-frequency identification (RFID) [1], which is a technology that identifies and tracks objects, such as a product, an animal or a person by using radio waves to transfer data from an electronic tag, called the RFID tag. In the last decade, a large volume of research was conducted on complying RFID systems with the international standard ISO/IEC 18000-7. The AMQM system exhibits some features of the ISO/IEC 18000-7 standard and hence its formal analysis is quite important.

In an AMQM system, tags are intended for identifying the objects that are to be managed. The interrogator communicates with the tag in its RF (Radio Frequency) communication range and controls the protocol, reads information from the tag, directs the tag to store data in some cases, and makes sure that messages are delivered and are also valid. An interrogator controls the messages that are transmitted during their allotted time periods called slots and an acknowledge that is received when each message has been received successfully.

Based on the AMQM communication protocol, the timing diagram of a tag collection process is depicted in Figure. 3.3. Thereafter, the communication sequence starts with a Wakeup Period (WP), within which wake up signals are sent to bring

all tags in the ready state. The WP is followed by a collection round named Command Period (CP), which in turn consists of a collection command period, a Listen Period (LP) and an Acknowledge Period (AP). The interrogator then waits for the responses from the tags that are sent randomly. The tag collection is done based on a predetermined algorithm that complies with the ISO/IEC 18000 7 standard. Thus, this system has two properties:

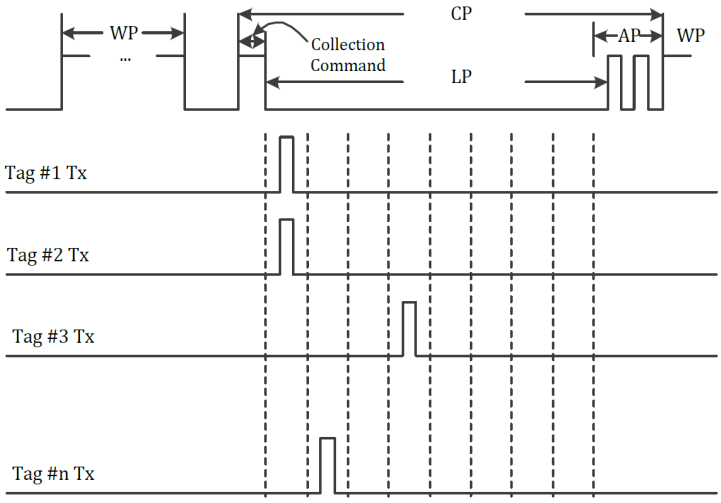

Figure 3.3: Tag Collection Process

1. The probability that a message can be delivered successfully within i slots is 1 - $(\frac{n-1}{n})^i$.

2. If the collection process is long enough, eventually any message can be delivered successfully.

This communication protocol can be modeled as a DTMC with 4 states: s_0(start), s_1(try), s_2(lost) and s_3(delivered) [1], as shown in Figure 3.4.

In the *start* state, the message is generated. The next state is always the state *try* and thus the probability from the *start* state to the *try* state is 1. The probability

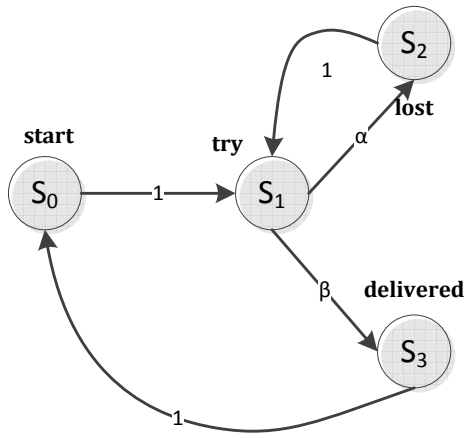

Figure 3.4: DTMC Model of the AMQM Protocol

of losing a message is α. Thus in the case of losing a message, the system will move to the *lost* state with probability α. Whereas, it moves to the *delivered* state with probability $\beta = 1 - \alpha$ in case of a successful transmission. Hence, the probability that a message can be delivered successfully is β, which equals to 1 - α. Once a message is delivered successfully, the system moves to the *start* state for getting ready to identify the other tags in the next time slot. When the collection process ends, the system goes to sleep mode in order to minimize power consumption. The state transition probability matrix, corresponding to the Markov chain given in Figure 3.4, is as follows:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1-1/n & 1/n \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}; I = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.6}$$

The initial distribution and transition probability can be expressed as two functions in higher-order logic as follows:

Definition 3.13. *(Initial Distribution and Transition Probability)*

```
⊢ Li i  = if (i = 0) then 1 else 0

⊢ Lt n t i j = case (i,j) of
    (0, 1) → 1 | (0, _) → 0 | (1, 2) → 1 - 1/n | (1, 3) → 1/n |
    (1, _) → 0 | (2, 1) → 1 | (2, _) → 0 | (3, 0) → 1 | (3, _) → 0
```

Now, the tag collection process involved in the AMQM protocol can be formalized as a time-homogeneous DTMC, based on the state diagram and the initial distribution and transition probability matrix.

Definition 3.14. *(AMQM Collection Model)*

```
⊢ ∀ X p a b c d.  AMQM_MODEL X p n =
   th_dtmc X p ([0, 3], POW [0, 3]) Li (Ltr n) ∧ (1 ≤ n)
```

Here, the state space is represented as a pair, in which the set `[0, 3]` contains all the states and `POW [0,3]` is the sigma-algebra of the states set `[0, 3]`, to ensure this state space is measurable. `Li` and `(Ltr n)` are the initial distribution and the transition probability matrix described in Definition 3.6. Variable `n` denotes the number of tags that are sent randomly. The first condition ensures the model to be a time-homogeneous DTMC. The second condition interprets that more than one node is considered in this collection process. Generally, the possible path of delivering a message successfully can be expressed as $\pi = (start, try, (lost, try)^k, delivered)$.

Here, k represents the number of iterations required for a successful message transmission. We use $\mathcal{P}_r(\diamond delivered_i)$ to represent the probability of delivering a message within i trials, where $\diamond$ is a logic symbol and means that "eventually" the event $delivered_i$ happens. Then the probability of reaching state s_3 is given by the

following equation where n represents the number of tags.

$$\mathcal{P}_r(\diamond delivered_i) = \sum_{k=0}^{i-1} \alpha^k \beta = 1 - (\frac{n-1}{n})^i \tag{3.7}$$

If the collection process is long enough, that is i tends to $+\infty$, then finally the message always can be delivered successfully. So the probability of delivering a message successfully in the future is

$$\mathcal{P}_r(\diamond delivered) = \sum_{k=0}^{\infty} \alpha^k \beta = \frac{\beta}{1-\alpha} = \frac{\frac{1}{n}}{1-\frac{n-1}{n}} = 1 \tag{3.8}$$

As mentioned before, the probability of reaching the *delivered* state depends on the tag collection algorithms, for example, in [1], an improved algorithm is presented for fast tag collection. Thus, Equations (3.7) and (3.8) play a vital role in assessing the performance of a tag collection algorithm. In this thesis, we formally verify these equations and our results can in turn be used to formally reason about the effectiveness of a tag collection algorithm.

Now, the two properties presented in Equations (3.7) and (3.8) can be expressed as the goals in the following two theorems:

Theorem 3.12. *(Probability of Reaching Delivered State in AMQM Protocol Model)*

$$\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ n\ i}.\\
&\quad \mathtt{AMQM_MODEL\ X\ p\ n} \Rightarrow\\
&\quad \mathtt{sum\ (0,\ i)\ (\lambda\ k.}\ \ \mathbb{P}(\{\mathtt{X}_{2+k*2}\ \mathtt{=}\ 3\} \cap \bigcap_{m=0}^{k-1}(\{\mathtt{X}_{3+m*2}\ \mathtt{=}\ 1\} \cap \{\mathtt{X}_{2+m*2}\ \mathtt{=}\ 2\}) \cap\\
&\qquad\qquad \{\mathtt{X}_1 = 1\} \cap \{\mathtt{X}_0 = 0\}))\ \mathtt{=}\ 1\ \mathtt{-}\ (\tfrac{n-1}{n})^i
\end{aligned}$$

Theorem 3.13. *(Reachability Probability of AMQM Protocol)*

$$\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ n}.\\
&\quad \mathtt{AMQM_MODEL\ X\ p\ n} \Rightarrow
\end{aligned}$$

$$\texttt{lim}\ (\lambda\ \texttt{i}.\ \texttt{sum}\ (0,\ \texttt{i})\ (\lambda\ \texttt{k}.\ \mathbb{P}(\{\texttt{X}_{2+\texttt{k}*2}\ =\ 3\}\ \cap\ \bigcap_{\texttt{m}=0}^{\texttt{k}-1}(\{\texttt{X}_{3+\texttt{m}*2}\ =\ 1\}\ \cap\ \{\texttt{X}_{2+\texttt{m}*2}\ =\ 2\})\ \cap\ \{\texttt{X}_1 = 1\}\ \cap\ \{\texttt{X}_0 = 0\}))) = 1$$

Proof. The proofs of both Theorems 3.12 and 3.13 are mainly based on Theorem 3.1 along with some arithmetic reasoning.

Theorem 3.12 corresponds to Equation (3.7), in which `i` refers to the number of trials required for successfully delivering `n` tags. The condition $n \neq 0$ means that the system will not be waken up if no tag is detected. The performance of a tag collection algorithm can be evaluated by this probability. Theorem 3.13 verifies that the probability of reaching the *delivered* state in infinite trials is 1. That is to say, if the tag collection process is long enough, at last all the tags generated at *start* state will be received by the reader successfully.

In [87], the PRISM model checker [94] has been used to analyze the AMQM protocol described above. To verify its correctness, the property expressed in Theorem 3.13 was verified from the point of view of reaching a good state in [87]. The verification of this property is based on solving a group of linear equations instead of verifying a PCTL expression mainly because this property involves an infinite summation, which is impossible to express in PCTL. Similarly, the collision probabilities, corresponding to Equation (3.7), have been verified for some special cases using iterative algorithms. Due to the inherent nature of numerical methods based analysis, these analyses cannot be termed accurate despite consuming enormous computing resources. Moreover, these results are not generic like the ones reported in Theorem 3.12 of this thesis, which means that the complete analysis has to be redone in case the information about the number of tags or time slots changes. On the other hand,

the proposed theorem proving based approach allowed us to formally reason about the generic expressions of two of the most important characteristics of the AMQM protocol, namely, the probability of reaching delivered state in the AMQM protocol model and the reachability probability of AMQM protocol. The results exactly match the results obtained via paper-and-pencil proof methods [87].

3.6 Summary and Discussion

In this chapter, we present a higher-order-logic formalization of the DTMC with a finite state space, which can be regarded as the first step towards a successful theorem-proving based analysis of DTMC. This formalization allows to model polymorphic states in different systems, where the states are not required to be positive integer. Another flexibility of this definition is that both time-homogeneous and time-inhomogeneous DTMC can be modeled based on this formalization. When the transition probabilities are time dependant, then the parameter p_{ij}, contains the dependence on time t. Furthermore, the general state space $\mathtt{s}$ in Definition 3.3 covers two cases: infinite or finite state spaces. Since the time-homogeneous DTMCs are the most frequently used in many applications, we present the definition of time-homogeneous DTMC.

Building upon the formalization of DTMC, the most important theorems, such as *joint probability theorem*, *Chapman-Kolmogorov Equation* and *Absolute probability*, were formally verified in HOL. These theorems are foundational for analyzing all kinds of DTMC models. Then, we defined the reversible stochastic process and detailed balance equations, which facilitate the verification of the reversibility property of a DTMC. Also, we described the formalizations of stationary distribution and stationary process, which are the basic concepts of the performance analysis. Using these notions,

we verified some interesting properties of the DTMC that satisfy the detailed balance equations.

Furthermore, we can formally define *mixing time* and *coupling* [67] in HOL based on the definition of stationary distribution presented in Section 3.3 and verify the relevant theorems, like *rapid mixing property* and *Convergence Theorem* [67]. One of the well-known ferromagnetism statistical models, *Ising model* [67], can be formalized based on the Markov chain mixing time notations and it can be applied in neuroscience [48], physics and statistical genetics [73]. On the other hand, our formalization of DTMC is general enough to describe time-inhomogeneous DTMCs, which are part of the notations in MCMC algorithms [11] and are also applied in many fields, like the *adiabatic theorem* [60] in quantum computations, by means of instantiating the transition probability (in Definition 3.3), which is a function taking the time t as its argument.

The main challenge of our work is to describe the Markov property and a DTMC using the proper and flexible predicates, respectively, in the higher-order logic. The proof script of the formalization and verification of the notions presented in this chapter require around 2000 lines of HOL4 code [68].

The binary communication channel is a typical DTMC model and many telecommunication systems are based on this basic structure. For this reason, we analyzed some properties of this basic channel structure as the first application presented in this chapter. As the first step, this channel is formalized as a DTMC model using higher-order logic. Then, two interesting properties of this channel are proved based on this model. The proof script only requires around 500 lines of HOL4 code. This example mainly illustrates a flow of the complete verification process of a DTMC model using theorem proving and it shows the usefulness of our formalization of DTMC.

The DTMC presented in the above example can also been interpreted as the time-dimension model in a Dynamic Spectrum Access/Cognitive Radio (DSA/CR) system [111], which is constructed for measuring spectrum occupancy, or the s-dependent Bernoulli trials in software reliability models [29], which is used to estimating the software failures. Also, the simple random walk with no barriers [3] can also be expressed using this DTMC model.

Another interesting application presented in this chapter is the verification of two properties of the tag collection process in the AMQM protocol. This process is formally defined as a time-homogeneous DTMC model based on the state diagram described in the specification. Usually, people are interested in learning the probability of all the tags sent out by the nodes involved in a wireless network are delivered successfully. We proved theorems presented in Section 3.2 to verify the properties of this DTMC model with around 600 lines of code in HOL.

In the telecommunications domain, various systems, such as the time elapsing in a synchronous fashion of a Bluetooth device [20], wireless LAN [63] and broadcast protocols [22], are described using state diagrams, which can be modeled as the DTMCs. The existing library of the formalized DTMC presented in this chapter can be applied to verify these telecommunication system models as well as those in many other domains.

The formalization of DTMC and the verification of the properties presented in this chapter, are the fundamental notations in DTMC theory, which facilitate the analysis of classical properties of general Markovian models. In fact, distinct discrete-time Markov chains exhibit diverse attractive features in analyzing long-term properties. These interesting characteristics can be analyzed based on classified states and classified DTMCs, which will be elaborated in the next chapter.

Chapter 4

Classified Discrete-Time Markov Chain in HOL

In this Chapter, the formalization of classified DTMCs will be introduced. We first formalize some foundational notions of classified states, which are categorized based on reachability, periodicity or absorbing features. Then, these results along with our formal definition of a DTMC are used to formalize classified Markov chains, such as aperiodic and irreducible DTMCs. Based on these concepts, some long-term properties are verified for the purpose of formally checking the correctness of the functions of Markovian systems or analyzing the performance of Markov chain models.

4.1 Classified States

The foremost concept of states classification is the *first passage time* τ_j, or the *first hitting time*, which is defined as the minimum time required to reach a state j from

the initial state i:

$$\tau_j = min\{t > 0 : X_t = j\}.$$

The first passage time can be defined in HOL as:

Definition 4.1. *(First Passage Time)*

```
⊢ ∀ X x j.  FPT X x j = MINSET {t | 0 < t ∧ (X t x = j)}
```

where `X` is a random process and `x` is a sample in the probability space associated with the random variable X_t. Note that the first passage time is also a random variable.

The conditional distribution of τ_j defined as the probability of the events starting from state i and visiting state j at time n is expressed as $f_{ij}^{(n)} = \mathcal{P}r\{\tau_j = n | X_0 = i\}$. This definition can be formalized in HOL as follows:

Definition 4.2. *(Probability of First Passage Events)*

```
⊢ ∀ X p i j n.
    f X p i j n = ℙ({x | FPT X x j = n}|{x | X 0 x = i})
```

Another important notion, denoted as f_{ij}, is the probability of the events starting from state i and visiting state j at all times n, is expressed as $f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)}$. It can be expressed in HOL as `(λ n.  f X p i j n) sums` f_{ij}. Thus f_{jj} provides the probability of events starting from state j and eventually returning back to j. If $f_{jj} = 1$, then the *mean return time* of state j is defined as $\mu_j = \sum_{n=1}^{\infty} n f_{jj}^{(n)}$. The existence of this infinite summation can be specified as `summmable (λ n.  n * f X p j j n)` in HOL.

A state j in a DTMC $\{X_t\}_{t \geq 0}$ is called *transient* if $f_{jj} < 1$, and *persistent* if $f_{jj} = 1$. If the mean return time μ_j of a persistent state j is finite, then j is said to be *persistent nonnull state* (or *positive persistent state*). Similarly, if μ_j is infinite, then j is termed as *persistent null state*.

The greatest common divisor (*gcd*) of a set is a frequently used mathematical concept in defining classified states. We formalize the gcd of a set as follows:

Definition 4.3. *(The gcd of a Set)*

```
⊢ ∀ A. GCD_SET A = MAXSET {r | ∀ x.  x ∈ A ⇒ divides r x}
```

where `MAXSET` is a function in the Set Theory of HOL4 such that `MAXSET s` defines the maximum element in the set `s`. A *period* of a state j is any n such that $p_{jj}^{(n)}$ is greater than 0 and we write $d_j = gcd\ \{n : p_{jj}^{(n)} > 0\}$ as the gcd of the set of all periods.

A state i is said to be *accessible* from a state j (written $i \rightarrow j$), if there exists a nonzero n-step transition probability of the events from state i to j. Two states i and j are called *communicating states* (written $i \leftrightarrow j$) if they are mutually accessible. A state j is an *absorbing state* if the one-step transition probability $p_{jj} = 1$. The formalization of some other foundational notions of the classified states is given in Table 4.1.

4.2 Classified DTMCs

In this section, we build upon the above mentioned definitions to formalize classified DTMCs. Usually, a DTMC is said to be *irreducible* if every state in its state space can be reached from any other state including itself in finite steps.

Definition 4.4. *(Irreducible DTMC)*

```
⊢ ∀ X p s p0 pij .
  Irreducible_mc X p s p0 pij =
  th_dtmc X p s p0 pij ∧
  (∀ i j.  i ∈ space s ∧ j ∈ space s ⇒
           Communicating_states X p s i j)
```

Table 4.1: Formalization of Classified States

Definition	**Condition**	**HOL Formalization**
Transient State	$f_{jj} < 1$	⊢ ∀ X p j. Transient_state X p j = ∀ x. {t | 0 < t ∧ (X t x = j)} ≠ ∅ ∧ (∃ s. s < 1 ∧ (λ n. f X p j j n) sums s)
Persistent State	$f_{jj} = 1$	⊢ ∀ X p j. Persistent_state X p j = ∀ x. {t | 0 < t ∧ (X t x = j)} ≠ ∅ ∧ (λ n. f X p j j n) sums 1
Persistent Nonnull State	$f_{jj} = 1$ $\mu_j < \infty$	⊢ ∀ X p j. Nonnull_state X p j = Persistent_state X p j ∧ summable (λ n. n * f X p j j n)
Persistent Null State	$f_{jj} = 1$ $\mu_j = \infty$	⊢ ∀ X p j. Null_state X p j = Persistent_state X p j ∧ ∼ summable (λ n. n * f X p j j n)
Periods of a State	0 < n $0 < p_{jj}^n$	⊢ ∀ X p s j. Period_set X p s j = {n | 0 < n ∧ ∀ t. 0 < Trans X p s t n j j}
GCD of a Period Set	d_j	⊢ ∀ X p s j. Period X p s j = GCD_SET (Period_set X p s j)
Periodic State	$d_j > 1$	⊢ ∀ X p s j. Periodic_state X p s j = (1 < Period X p s j) ∧ (Period_set X p s j ≠ ∅)
Aperiodic State	$d_j = 1$	⊢ ∀ X p s j. Aperiodic_state X p s j = (Period X p s j = 1) ∧ (Period_set X p s j ≠ ∅)
Accessibility	i → j	⊢ ∀ X p s i j. Accessibility X p s i j = ∀ t. ∃ n. 0 < n ∧ 0 < Trans X p s t n i j
Communicating State	i ↔ j	⊢ ∀ X p s i j. Communicating_states X p s i j = (Accessibility X p s i j) ∧ (Accessibility X p s j i)
Absorbing State	$p_{jj} = 1$	⊢ ∀ X p s j. Absorbing_states X p s j = (Trans X p s t 1 j j = 1)

where th_dtmc is time-homogeneous Markov chain defined in Definition 3.4 and the second conjunct expresses that all the states in the state space can communicate with each other.

If there exists a state in the state space of a DTMC, which cannot reach some other states, then this DTMC is called *reducible*.

Definition 4.5. *(Reducible DTMC)*

$$
\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ s}\ p_0\ p_{ij}\,.\\
&\quad \mathtt{Reducible\_mc\ X\ p\ s}\ p_0\ p_{ij}\ =\\
&\quad \mathtt{th\_dtmc\ X\ p\ s}\ p_0\ p_{ij}\ \wedge\\
&\quad \exists\ \mathtt{i\ j.}\ \ \mathtt{i} \in \mathtt{space\ s} \wedge \mathtt{j} \in \mathtt{space\ s} \wedge\\
&\qquad\quad \sim\mathtt{Communicating\_states\ X\ p\ s\ i\ j}
\end{aligned}
$$

A DTMC is considered as *aperiodic* if every state in its state space is an aperiodic state; otherwise it is a *periodic DTMC*.

Definition 4.6. *(Aperiodic DTMC)*

$$
\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ s}\ p_0\ p_{ij}\,.\\
&\quad \mathtt{Aperiodic\_mc\ X\ p\ s}\ p_0\ p_{ij}\ =\\
&\quad \mathtt{th\_dtmc\ X\ p\ s}\ p_0\ p_{ij}\ \wedge\\
&\quad (\forall\ \mathtt{i.}\ \ \mathtt{i} \in \mathtt{space\ s} \Rightarrow \mathtt{Aperiodic\_state\ X\ p\ s\ i})
\end{aligned}
$$

Definition 4.7. *(Periodic DTMC)*

$$
\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ s}\ p_0\ p_{ij}\,.\\
&\quad \mathtt{Periodic\_mc\ X\ p\ s}\ p_0\ p_{ij}\ =\\
&\quad \mathtt{th\_dtmc\ X\ p\ s}\ p_0\ p_{ij}\ \wedge\ (\exists\ \mathtt{i.}\ \ \mathtt{i} \in \mathtt{space\ s} \wedge \mathtt{Periodic\_state\ X\ p\ s\ i})
\end{aligned}
$$

If at least one absorbing state exists in a DTMC and it is possible to go to the absorbing state from every non-absorbing state, then such a DTMC is named as an *absorbing DTMC*.

Definition 4.8. *(Absorbing DTMC)*

$$\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ s\ p_0\ p_{ij}}.\\
&\quad \mathtt{Absorbing_mc\ X\ p\ s\ p_0\ p_{ij}} =\\
&\quad \mathtt{th_dtmc\ X\ p\ s\ p_0\ p_{ij}} \wedge\\
&\quad \exists\ \mathtt{i.}\ \ \mathtt{i} \in \mathtt{space\ s} \wedge \mathtt{Absorbing_state\ X\ p\ s\ i} \wedge\\
&\qquad (\forall\ \mathtt{j.}\ \ \mathtt{j} \in \mathtt{space\ s} \Rightarrow \mathtt{Communicating_state\ X\ p\ s\ i\ j})
\end{aligned}$$

Finally, if there exists some n such that $p_{ij}^{(n)} > 0$ for all states i and j in a DTMC, then this DTMC is defined as a *regular DTMC*.

Definition 4.9. *(Regular DTMC)*

$$\begin{aligned}
&\vdash \forall\ \mathtt{X\ p\ s\ p_0\ p_{ij}}.\\
&\quad \mathtt{Regular_mc\ X\ p\ s\ p_0\ p_{ij}} =\\
&\quad \mathtt{th_dtmc\ X\ p\ s\ p_0\ p_{ij}} \wedge\\
&\quad \exists\ \mathtt{n.}\ \ \forall\ \mathtt{i\ j.}\ \ \mathtt{i} \in \mathtt{space\ s} \wedge \mathtt{j} \in \mathtt{space\ s} \Rightarrow \mathtt{Trans\ X\ p\ s\ t\ n\ i\ j} > 0
\end{aligned}$$

The main utility of the higher-order logic formalization of the classified Markov chains mentioned above is to formally specify and analyze the dynamic features of Markovian systems within the sound environment of a theorem prover as will be demonstrated in the following section.

4.3 Long-term Properties

The analysis of the long-term behavior of a DTMC is dependent on the type of state under consideration. Among the classified DTMCs formalized in the previous section,

aperiodic and irreducible DTMCs are considered to be the most widely used ones in analyzing Markovian systems because of their attractive stationary properties, i.e., their limit probability distributions are independent of the initial distributions. For this reason, we now focus on the verification of some key properties of aperiodic and irreducible DTMCs [34].

4.3.1 Positive Transition Probability

Based on the the aperiodicity and irreducibility characteristics of certain DTMCs, we can prove the most important properties of them. One of the results is that aperiodic and irreducible DTMCs possess positive transition probabilities at some point, which is useful to prove the convergence theorems in next subsection. To prove this property, we have to start from some fundamental theorems.

Theorem 4.1. *(Closed Period Set)*

In an aperiodic DTMC, the set of the times when state i has a non-null probability of being visited is closed under addition.

```
⊢ ∀ X p s p₀ pᵢⱼ i.
   Aperiodic_DTMC X p s p₀ pᵢⱼ ∧ i ∈ space s ⇒
   ∀ a b.  a ∈ Period_set X p s i ∧ b ∈ Period_set X p s i ⇒
        (a + b) ∈ Period_set X p s i
```

Proof. We verified the above theorem by rewriting the goal with the definition of `Period_set` given in Table 4.1 and Definition 4.6, and then applying Theorem 3.2 along with some arithmetic and set theoretic reasoning.

Another key property of an aperiodic DTMC indicates that the transition probability $p_{ij}^{(n)}$ is greater than zero, for all states i and j in its state space. It is very

useful in analyzing the stability or reliability of real-world systems.

Theorem 4.2. *(Positive Return Probability)*

For any state i in the finite state space S of an aperiodic DTMC, there exists an $N < \infty$ such that $0 < p_{ii}^{(n)}$, for all $n \geq N$ and all i in the state space.

```
⊢ ∀ X p s p0 pii i t.
   Aperiodic_DTMC X p s p0 pii ∧ i ∈ space s ⇒
   ∃ N. ∀ n.  N ≤ n ⇒ 0 < Trans X p s t n i i
```

In this theorem, `N` refers to a nature number and its type is `num`. The variable with this kind of type is less than infinity in HOL. The formal reasoning about the correctness of the above theorems involves Theorems 3.2 and 4.1 and the following Lemmas 4.1, along with some arithmetic reasoning and set theoretic reasoning.

Lemma 4.1. *(Positive Element in a Closed Set)*

If an integer set S contains at least one nonzero element and S is closed under addition and subtraction, then $S = \{kc; k \in \mathbb{Z}\}$, where c is the least positive element of S.

```
⊢ ∀ s:int → bool.  s ≠ ∅ ∧
   (∀ a b.  a ∈ s ∧ b ∈ s ⇒ (a + b) ∈ s ∧ (a - b) ∈ s) ⇒
   0 < MINSET {r | 0 < r ∧ r ∈ s} ∧
   (s = {r | ?k.  r = k * MINSET {r | 0 < r ∧ r ∈ s}})
```

where `MINSET A` refers to the minimum element in the set `A` and the type of `A` is integer.

Lemma 4.2. *(Linearity of Two Integer Sequences)*

For a positive integer sequence a_1, a_2, $\cdots$, a_k, there exists an integer sequence n_1, n_2, $\cdots$, n_k, such that $d = \sum_{i=1}^{k} n_i a_i$, where d is the greatest common divisor of sequence a_1, a_2, $\cdots$, a_k.

```
⊢ ∀ a k.  0 < k ∧ (∀ i.  i ≤ k ⇒ 0 < a i) ⇒
  (∃ n.  GCD_SET {a i | i ∈ [0, k]} = ∑k i=0 n i * a i)
```

Lemma 4.3. *(Least Number)*

If a set of positive integers A is nonlattice, i.e., its gcd is 1, and closed under addition, then there exists an integer $N < \infty$ such that $n \in A$ for all $N \leq n$.

```
⊢ ∀ (A:int → bool) a.
  (A = {a i | 0 < a i ∧ i ∈ UNIV(:num)}) ∧ (GCD_SET A = 1) ∧
  (∀ a b.  a ∈ A ∧ b ∈ A ⇒ (a + b) ∈ s) ⇒ (∃ N. {n | N ≤ n} ⊂ A)
```

The proofs of Lemmas 4.1, 4.2 and 4.3 are based upon various summation properties of integer sets and the properties of gcd of a set. These properties were not available in the HOL libraries and thus had to be verified as part of our development. The detailed proof steps can be found in [11] and the proof script for these lemmas, including their prerequisite results are available at [68].

Theorem 4.3. *(Existence of Positive Transition Probabilities)*

For any aperiodic and irreducible DTMC with finite state space S, there exists an N, such that, for all $n \geq N$, the n-step transition probability $p_{ij}^{(n)}$ is non-zero, for all states i and $j \in S$.

```
⊢ ∀ X p s p₀ pᵢⱼ i j t.
   Aperiodic_DTMC X p s p₀ pᵢⱼ ∧ Irreducible_DTMC X p s p₀ pᵢⱼ ∧
   i ∈ space s ∧ j ∈ space s ⇒
   ∃ N. ∀ n.  N ≤ n ⇒ 0 < Trans X p s t n i j
```

Proof. We proceed with the proof of Theorem 4.3 by performing case analysis on the equality of i and j. The rest of the proof is primarily based on Theorems 3.2 and 4.2, Definition 3.1 and Lemmas 4.2 and 4.3.

Theorem 4.4. *(Existence of Long-run Transition Probabilities)*

For any aperiodic and irreducible DTMC with finite state space S and transition probabilities p_{ij}, there exists $\lim_{n\to\infty} p_{ij}^{(n)}$, for all states i and $j \in S$.

```
⊢ ∀ X p s p₀ pᵢⱼ i j t.
   Aperiodic_DTMC X p s p₀ pᵢⱼ ∧ Irreducible_DTMC X p s p₀ pᵢⱼ ⇒
   ∃ u.  lim (λ n.  Trans X p s t n i j) = u
```

Proof. We first prove the monotonic properties of M_j^n and m_j^n, which are the maximum and minimum values of the set $\{n \geq 1\text{: } p_{ij}^{(n)}>0\}$, respectively. Then, the proof is completed by verifying the convergence of the sequence (M_j^n - m_j^n) for all n such that $N \leq n$ by applying Theorem 3.2 and some properties of real sequences. It is important to note that we do not need to use the assumption $j \in$ *space* s here, like all other theorems, as $\forall\ n\ j.\ j \notin$ *space* $s \Rightarrow (p_j^{(n)} = 0)$, which in turn implies $\lim_{n\to\infty} p_j^{(n)} = 0$ and $\lim_{n\to\infty} p_{ij}^{(n)} = 0$ by using the conditional probability theorem presented in Equation (2.3a).

4.3.2 Convergence Analysis

The long-run probability distributions are often considered in the convergence analysis of random variables in stochastic systems. It is not very easy to verify that the limit probability distribution of a certain state exists in a generic non-trivial DTMC, because the computations required in such an analysis are often tremendous. However, in the aperiodic and irreducible DTMCs, we can prove that all states possess limiting probability distributions, by the following two theorems.

Theorem 4.5. *(Existence of Long-run Probability Distributions)*

For any aperiodic and irreducible DTMC with finite state space S, there exists $\lim_{n\to\infty} p_i^{(n)}$*, for any state* $i \in S$.

```
⊢ ∀ X p s p₀ pᵢⱼ i.
   Aperiodic_DTMC X p s p₀ pᵢⱼ ∧ Irreducible_DTMC X p s p₀ pᵢⱼ ⇒
   ∃ u.  (λ n.  ℙ{x | X n x = i} → u)
```

Proof. We used Theorems 3.3 and 4.4, along with some properties of the limit of a sequence, to prove this theorem in HOL.

Theorem 4.6. *(Existence of Steady State Probability)*

For every state i in an aperiodic and irreducible DTMC, $\lim_{n\to\infty} p_i^{(n)}$ *is a stationary distribution.*

```
⊢ ∀ X p s p₀ pᵢⱼ.
   Aperiodic_DTMC X p s p₀ pᵢⱼ ∧ Irreducible_DTMC X p s p₀ pᵢⱼ ⇒
    (stationary_dist p X (λ i.  lim (λ n.  ℙ{x | X n x = i}) s)
```

Proof. The proof of Theorem 4.6 involves rewriting with Definition 3.8 and then splitting it into the following three subgoals:

- $0 \leq \lim_{n\to\infty} p_j^{(n)}$
- $\sum_{i\in\Omega} \lim_{n\to\infty} p_i^{(n)} = 1$
- $\lim_{n\to\infty} p_j^{(n)} = \sum_{i\in\Omega} \lim_{n\to\infty} p_i^{(n)} p_{ij}$

Utilizing the probability bounds theorem, we can prove the first subgoal. The proof of the second subgoal is primarily based on the additivity property of conditional probability [41]. Then the last subgoal can be proved by applying the linearity of limit of a sequence and the linearity of real summation.

4.4 Applications

In the previous section, the most important theorems of classified DTMCs are provided in higher-order logic to analyze the long-term behavior of Markovian systems. In order to demonstrate their usefulness, in this section, we first present a formal validation of a LRU stack Model in HOL4. Then, we utilize the formalization of aperiodic and irreducible DTMCs to formally define a discrete-time Birth-Death Process [109], which can be applied in formally analyzing the performance of software data structure.

4.4.1 LRU Stack Model

In a Least Recently Used (LRU) stack model, as shown in Figure 4.1, a sequence of stacks $s_1 s_2 \cdots s_t \cdots$ are associated with a reference string $w = \mathrm{x}_1\mathrm{x}_2\cdots\mathrm{x}_t\cdots$. Any stack s_t is a n-tuple $(j_1, j_2, \cdots, j_n)$, where j_i refers to the i^{th} most recently referenced page at time t [109]. Let D_t be the position of the page x_t in the stack s_{t-1}. Then the distance string is $\mathrm{D}_1\mathrm{D}_2\cdots\mathrm{D}_t\cdots$, which is associated with the referencing string. This distance string can be modeled as a sequence of independent and identically

distributed (IID) random variables [4], which makes their probability mass function (PMF) as $\mathcal{P}r(D_t = i) = \mathrm{a}_i$, where $i = 1, 2, \ldots, n$ and refers to the position of the least recently used page in the stack at time t, and $\sum_{j=1}^{n} \mathrm{a}_j = 1$. This way the distribution function becomes $\mathcal{P}r(D_t \leq i) = \sum_{j=1}^{i} \mathrm{a}_j$. Now, if a tagged page occupies the i^{th} position in the stack at time t, which is expressed as s_t in Figure 4.1, then the position of this page in the stack s_{t+1} depends on the next reference x_{t+1} and the position of this page in the stack s_t.

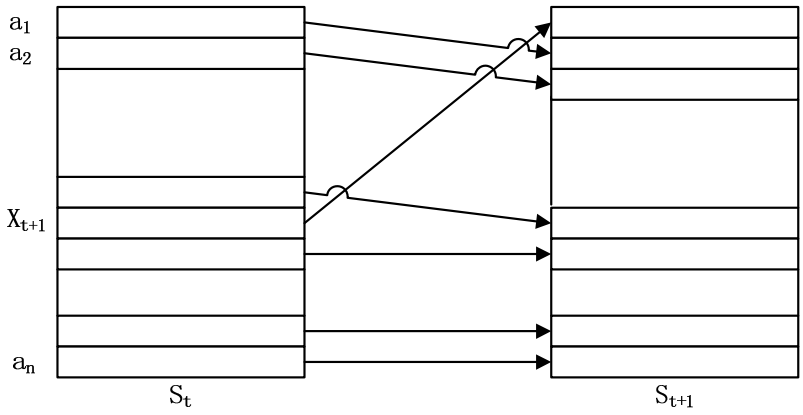

Figure 4.1: LRU Stack Updating Procedure

Based on the described updating procedure in the LRU stack, the evaluation of the page-fault rate of the LRU paging algorithm becomes quite simple. If the evaluated program has been allocated i page frames of main memory, then a page fault will occur at time t when $D_t > i$. Hence, the page fault probability is [109]

$$\mathcal{F}(LRU) = \mathcal{P}r(D_t > i) = 1 - \sum_{j=1}^{i} a_j \tag{4.1}$$

The movement of the tagged page through the LRU state is then a random process $\{E_t\}_{t \geq 0}$. If the page occupies the i^{th} position in the stack s_t, then $E_t = i$, for all i, $1 \leq i \leq n$. Now, we have the following transition probabilities:

$$p_{i1} = \mathcal{P}r(E_{t+1} = 1 | E_t = i) = \mathcal{P}r(D_{t+1} = i) = a_i, 1 \leq i \leq n$$

$$p_{ii} = \mathcal{P}r(E_{t+1} = i | E_t = i) = \mathcal{P}r(D_{t+1} < i) = \sum_{j=1}^{i-1} a_{j-1}, 2 \leq i \leq n$$

$$p_{i,i+1} = \mathcal{P}r(E_{t+1} = i+1 | E_t = i) = \mathcal{P}r(D_{t+1} > i) = 1 - \sum_{j=1}^{i} a_{j-1}, 1 \leq i \leq n-1$$

$$p_{i,j} = 0, otherwise.$$

The LRU stack is then described as an aperiodic and irreducible DTMC [109] by assuming $a_i > 0$ for all $i \in [1, n]$.

The state diagram of this aperiodic and irreducible DTMC is shown in Figure 4.2,

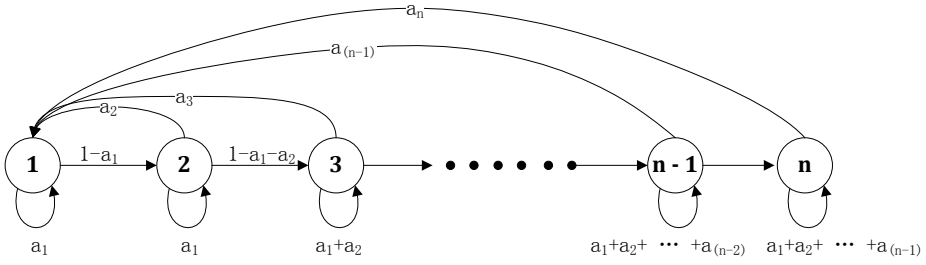

Figure 4.2: State Diagram for the LRU Stack Model

where we can find that the transition probabilities can be expressed as the following higher-order logic function [109]:

Definition 4.10. *(Transition Probability Matrix)*

```
⊢ Lt a t i j =
   if (j = 1) then a i else
   if (j - i = 1) then 1 - sum (1, i) (λ j.  a j) else
   if (j = i) then sum (1, i - 1) (λ j.  a j) else 0
```

which can be used to formalize the LRU stack model as:

Definition 4.11. *(LRU Model)*

```
⊢ LRU_model X p a n p0.
  Aperiodic_DTMC X p ([1, n], POW ([1, n])) p0 (Lt a) ∧
  Irreducible_DTMC X p ([1, n], POW ([1, n])) p0 (Lt a) ∧
  1 ≤ n ∧ (∀ j.  0 < j ∧ j ≤ n ⇒ 0 < a j) ∧
  (sum (1, n) (λ j.  a j) = 1)
```

where the state space is described as a pair `([1, n], POW ([1, n]))`, in which the first element contains all the states $\{1, 2, \cdots, n\}$ and the second one is the sigma algebra of the first element. The condition $(1 \leq n)$ is used to avoid the case when the length of the referencing string is zero. The other two conditions represent the specification of the model mentioned above.

Using the formal definition of this LRU stack model, we can now formally reason about its limiting distributions, which are mainly used to describe the stationary behaviors of this model.

Theorem 4.7. *(Existence of the Limiting State Distribution in the LRU Stack Model)*

In the LRU stack model, there exists $\lim_{t\to\infty} p_i^{(n)}$*, for every* $i \in [1, n]$*.*

```
⊢ ∀ X p a n p0 i.
  LRU_model X p a n p0 ∧ i ∈ [1, n] ⇒
  ∃ u.  (λ t.  ℙ{x | X t x = i} → u)
```

Proof. We verify this property by directly applying Theorem 4.5 and the definition of limit of a real sequence.

Theorem 4.8. *(LRU Stationary Limiting State Distribution)*

In the LRU stack model, $\lim_{t\to\infty} p_i^{(n)} = \frac{1}{n}$*, for every* $i \in [1, n]$*.*

$\vdash \forall$ `X p a n` p_0 `i.`

`LRU_model X p a n` $p_0 \wedge$ `i` $\in$ `[1, n]` $\Rightarrow$

`(lim (`λ `t. ` $\mathbb{P}${`x | X t x = i`}`) = `$\frac{1}{n}$`)`

Proof. The proof of this property is primarily based on Theorems 3.3 and 4.8 along with the following lemma:

Lemma 4.4. *(Identity Limiting State Distribution)*

$\vdash \forall$ `X p a n` p_0 `i j.`

`LRU_model X p a n` $p_0 \wedge$ `i` $\in$ `[1, n]` $\wedge$ `j` $\in$ `[1, n]` $\Rightarrow$

`(lim (`λ `t. ` $\mathbb{P}${`x | X t x = i`}`) = lim (`λ `t. ` $\mathbb{P}${`x | X t x = j`}`))`

Proof. The HOL proof of the above lemma is based on Theorem 3.3 along with some arithmetic reasoning.

Theorem 4.8 implies that $\lim_{t\to\infty} p_i^{(t)}$ (for any tag i) is independent of its initial distribution and the position of the tagged page has an equal probability to be in any stack position. This means that any page is equally likely to be referenced in the long run. As a result, it concludes that this LRU stack specification does not cover the case of nonuniform page referencing behaviors of some programs. Thus, we have been able to formally verify the numerical methods result presented in [106].

The ability to formally verify theorems involving classified Markovian models and the proof script only consists of about 400 lines code in HOL4. The short script clearly indicates the usefulness of the formalization, presented in the earlier section of this thesis, as without them the reasoning could not have been done in such a straightforward manner.

4.4.2 Discrete-time Birth-Death Process

The Birth-Death process is an important sub-class of Markov chains as it involves a state space with non-negative integers. Its remarkable feature is that all one-step transitions lead only to the nearest neighbor state. The discrete-time Birth-Death Processes are mainly used in analyzing software stability, for example, verifying if a data structure will have overflow problems.

The discrete-time Birth-Death Process can be described as a state diagram depicted in Figure 4.3.

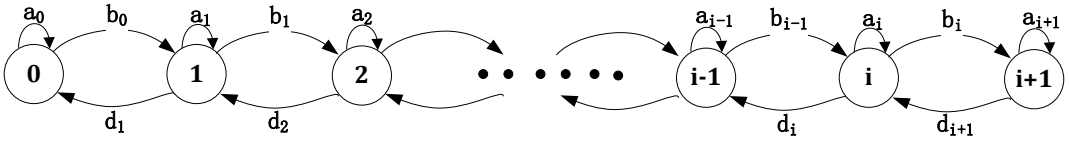

Figure 4.3: State Diagram of Discrete-time Birth-Death Process

In the above diagram, the states 0, 1, $\cdots$, i, $\cdots$ are associated with the population. The transition probabilities b_i represents the probability of a birth when the population is i, d_i denotes the probability of a death when the population becomes i, and a_i refers to the population in the state i.

Considering $0 \leq a_i \leq 1$, $< b_i < 1$ and $0 < d_i < 1$ (for all i, $1 \leq i \leq n$), the Birth-Death process described here is not a pure birth or pure death process as the population is finite. Thus, the Birth-Death process can be modeled as an aperiodic and irreducible DTMC [109].

In this DTMC model, the amount of population, usually considered, is greater than 1. Also, a_i, b_i and d_i should satisfy the additivity of probability axiom [109]. Then, the transition probability is given in the following function:

Definition 4.12. *(Transition Probability of Discrete-time Birth_Death Process)*

```
⊢ ∀ a b d t i j.
  DBLt a b d t i j =
   if (i = 0) ∧ (j = 0) then a 0 else
   if (i = 0) ∧ (j = 1) then b 0 else
   if (0 < i) ∧ (i - j = 1) then d i else
   if (0 < i) ∧ (i = j) then a i else
   if (0 < i) ∧ (j - i = 1) then b i else 0;
```

Based on these concepts, the discrete-time Birth-Death process is formalized as:

Definition 4.13. *(Discrete-time Birth_Death Process Model)*

```
⊢ ∀ X p a b c d n p₀.
   DB_MODEL X p a b d n p₀ =
   Aperiodic_MC X p ([0, n], POW [0,n]) p₀ (DBLt a b d) ∧
   Irreducible_MC X p ([0, n], POW [0,n]) p₀ (DBLt a b d) ∧
   1 < n ∧ (a₀ + b₀ = 1) ∧ ∀ j.  0 < j ∧ j < n ⇒ (aⱼ + bⱼ + dⱼ = 1) ∧
   ∀ j.  j < n ⇒ 0 < bⱼ ∧ bⱼ < 1 ∧ 0 < dⱼ ∧ dⱼ < 1
```

In this definition, this process is formally described as an aperiodic and irreducible DTMC, in which the state space is expressed as a pair `([0, n], POW [0,n])`. The set `[0, n]` represents the population and `POW [0,n]` is the sigma-algebra of the set `[0, n]`. Since the aperiodic and irreducible DTMC is independent of the initial distribution, the parameter $\mathtt{p_0}$ in this model is a general function. The other conjunctions shown in Definition 4.13 are the requirements described in the specification of the discrete-time Birth-Death process mentioned above.

Now, we can prove that this discrete-time Birth-Death process possesses the limiting probabilities.

Theorem 4.9. *(Discrete-time Birth-Death Process Exists Limit Probability)*

```
⊢ ∀ X p a b d n p₀ i.
   DB_MODEL X p a b d n p₀ ⇒ (∃ u.  ℙ{Xₜ = i} → u)
```

Proof. This theorem can be verified by rewriting the goal with Definition 4.13 and then applying Theorem 4.5.

Now, we can prove that the limit probability distributions are the stationary distributions, which are defined in Definition 3.8, and are independent of the initial probability vector as the following theorem.

Theorem 4.10. *(Stationary Distributions in a Discrete-time Birth-Death Process)*

```
⊢ ∀ X p a b d n p₀.
   DB_MODEL X p a b d n p₀ ⇒ (∃ f.  stationary_dist p X f s)
```

Proof. We prove this theorem by first instantiating `f` to be the limiting probabilities, `lim (λ t.  ℙ{Xₜ = i})`, and then by applying Theorem 4.9.

The last two theorems verify that the Birth-Death process holds the steady-state probability vector $v_i = \lim_{t \to \infty} \mathbb{P}\{X_t = i\}$. The computation of the steady-state probability vector v_i is mainly based on the following two Equations (4.2a) and (4.2b):

$$v_0 = a_0 v_0 + d_1 v_1 \tag{4.2a}$$

$$v_i = b_{i-1} v_{i-1} + a_i v_i + d_{i+1} v_{i+1} \tag{4.2b}$$

Now, these two equations can be formally verified by the following two theorems.

Theorem 4.11. *(Equation 4.2a)*

```
⊢ ∀ X p a b d n p₀.
   DB_MODEL X p a b d n p₀ ⇒
   (lim (λ t.  ℙ{Xₜ = 0}) =
     a₀ * lim (λ t.  ℙ{Xₜ = 0}) + d₁ * lim (λ t.  ℙ{Xₜ = 1}))
```

Proof. We first apply Theorems 3.3 and 4.10 to simplify the main goal, then finalize the proof by applying the conditional probability additivity theorem expressed in Equation (2.2), along with some arithmetic reasoning.

Theorem 4.12. *(Equation 4.2b)*

$\vdash \forall$ X p a b d n i p_0.

DB_MODEL X p a b d n p_0 $\wedge$ i + 1 $\in$ [0, n] $\wedge$ i - 1 $\in$ [0, n] $\Rightarrow$

(lim (λ t. $\mathbb{P}\{X_t = i\}$) =

b_{i-1} * lim (λ t. $\mathbb{P}\{X_t = i - 1\}$) + a_i * lim (λ t. $\mathbb{P}\{X_t = i\}$)

+ d_{i+1} * lim (λ t. $\mathbb{P}\{X_t = i + 1\}$))

Proof. We proceed the proof of this theorem by applying Theorems 3.3, 4.10, 4.11 and the Lemma given in Equation (2.3d), along with some arithmetic reasoning.

The general solution of the linear Equations (4.2a) and (4.2b) are expressed as:

$$v_{i+1} = \prod_{j=1}^{i+1} \frac{b_{j-1}}{d_j} v_0 \tag{4.3a}$$

$$v_0 = \frac{1}{\sum_{i=0}^{n} \prod_{j=1}^{i+1} \frac{b_{j-1}}{d_j}} \tag{4.3b}$$

These two equations are the major targets of the long-term behavior analysis and can be verified in higher-order logic as the following two theorems:

Theorem 4.13. *(Equation 4.3a)*

$\vdash \forall$ X p a b d n i Linit.

DB_MODEL X p a b d n Linit $\wedge$ i + 1 $\in$ [0, n] $\Rightarrow$

(lim (λ t. $\mathbb{P}\{X_t = i + 1\}$) =

lim (λ t. $\mathbb{P}\{X_t = 0\}$) * PROD (1, i + 1) (λ j. $\frac{b_{j-1}}{d_j}$))

Proof. The proof of this theorem starts with induction on the variable n. The base case can be verified by Theorem 4.11 and some real arithmetic reasoning. The proof

of the step case, then, is completed by applying a lemma which proves the following Equation (4.4) based on the DB_MODEL:

$$v_{i+1} = \frac{b_i}{d_{i+1}} v_{i+1} \tag{4.4}$$

The proof of this lemma is mainly done by induction on variable i. The base case is proved by applying Theorems 3.3, 4.10 and 4.11 as well as some real arithmetic reasoning. The proof of the step case is completed by using Theorem 4.12 along with some arithmetic reasoning.

Theorem 4.14. *(Equation 4.3b)*

$\vdash$ $\forall$ X p a b d n i Linit.

DB_MODEL X p a b d n Linit $\wedge$ i + 1 $\in$ [0, n] $\Rightarrow$

(lim (λ t. $\mathbb{P}\{\mathtt{X_t} = 0\}$) = $\frac{1}{\mathtt{sum}\ (0,\ n+1)\ (\lambda i.\ \mathtt{PROD}\ (1,\ i+1)\ (\lambda j.\ \frac{b_{j-1}}{d_j}))}$)

Proof. The proof of this theorem starts from rewriting the goal as lim (λ t. $\mathbb{P}\{\mathtt{X_t} = 0\}$) * sum $(0,\ n+1)$ ($\lambda i.$ PROD (1, i + 1) ($\lambda j.$ $\frac{b_{j-1}}{d_j}$) = 1. Then we split the summation into two items: $\frac{b_0}{d_1}$ and sum $(1,\ n+1)$ ($\lambda i.$ PROD (1, i + 1) ($\lambda j.$ $\frac{b_{j-1}}{d_j}$). The proof is completed by applying Theorems 4.11, 4.13 and the probability additivity theorem expressed in Equation (4.2a) and some real arithmetic reasoning.

After these theorems are verified, the limit probabilities of any states in this model can be calculated by instantiating the parameter n and transition probabilities a, b and d. Thus, it becomes unnecessary for the potential user to employ any numerical arithmetic to analyze the long-term behaviors of this model. The solution shown in Equations (4.3a) and (4.3b) is mainly used to predict safety properties in the development of the population in a long period, in various domains, such as statistics and biological.

More specifically, when the birth-death coefficients are $b_i = \lambda$ and $d_i = \mu$ (λ and μ are constants) for all i in the state space, then the model described in Definition

4.13 represents a classical M/M/1 queueing system [57] (in this case, the average inter arrival time becomes $\frac{1}{\lambda}$ and the average service time is $\frac{1}{\mu}$). Thereafter, the verified theorems can be directly applied to analyzing the ergodicity of M/M/1 queueing.

4.5 Summary and Discussion

This chapter first present a formal definition of the first passage time in higher-order logic. Based on this formalization, a series of classified states, including *transient state*, *persistent state* and *persistent null state*, as well as *nonnull persistent state*, are introduced. By introducing the formalization of the greatest common divisor (gcd) of a set, the periodic and aperiodic states are formally defined in HOL. We also present the higher-order-logic formalization of the communicating state and absorbing state. Building upon these definitions, we formalized a number of most common classified DTMCs. These concepts primarily appear in the reachability analysis or long-term behavior analysis.

To facilitate the probabilistic analysis of DTMC models, we verified the most important properties of aperiodic and irreducible DTMCs, which can be found in most textbooks and are frequently used in real-world applications. These properties (theorems in the higher-order logic) represent the foundation of classified DTMCs, which enables to derive more interesting properties of classified DTMCs. Moreover, these theorems are also frequently used in ergodic theory [55] due to the fact that aperiodic and irreducible DTMCs belong to the special class of ergodic systems. Furthermore, the properties of the ergodicity of DTMCs, the regular and absorbing DTMCs can be verified by applying their definitions, given in Section 4.2, and the theorems presented in Section 3.3. The absorbing DTMCs are frequently applied in modeling social-psychological problems [64]. Combining the formalization of absorbing DTMC

in Section 4.2 with the matrix theory [24], the properties of absorbing DTMC can also be formally verified using higher-order logic.

The proof scripts contain about 8000 lines of HOL4 code and are available at [68]. The major challenge of the work presented in this chapter is to find a way to formally verify the theorems. Most of the times, the detailed proof steps are not available in the textbooks. Moreover, some proofs in the textbook are doubtful. For instance, the proof steps of Lemma 4.3 presented in [11] are based on an underlying assumption that the order of the sequence $n_0, \cdots, n_i$ and $a_0, \cdots, a_i$ are correlated with each other, whereas this correlation property between these two sequences is not correct. Thus, most of the times we have to develop the proofs in HOL from scratch.

The LRU stack model is a typical model [99] for simulating the paging behaviors of memory management in computer architectures. Later, this model was found to be erroneous based on simulation results [110] and the authors commented that some experimental results had shown that the LRU stack depth distribution varies significantly among programs, even the same program running with different stimuli [110]. Furthermore, in [109], the LRU stack model is described as an aperiodic and irreducible DTMC, which can be validated by the developed formalization of classified DTMCs elaborated in this chapter. To proceed with the validation of this LRU stack model, we described it formally in HOL. Then we proved that this model exhibits the uniform page referencing behavior in the long-term, which exactly matched with the speculation stated in [110]. The proofs of this property of the LRU stack model require only 300 lines of code in HOL4. Compared with the large but limited experimental data, our formal validation of this model is more comprehensive and more accurate. This instance illustrates the usefulness of the approach proposed in this thesis and the flexibility of our higher-order logic formalization.

Another special type of DTMC, presented as an application in this chapter, is the discrete-time Birth-Death process. We modeled this process by applying the formalizations of aperiodic and irreducible DTMC and then verified the general equations ((4.3a) and (4.3b)) for expressing the limit probabilities of all states in the state space. Although the proof of these two equations required less than 400 lines of HOL4 code, these two theorems are very helpful on predicting the amount of the population considered in such kind of discrete-time Birth-Death process. Especially, if the discrete-time Birth-Death process model is used to represent the behaviors of a data structure which is being manipulated in a software program [109], then we can apply the theorems proved in Section 4.4.2 to analyze software reliability parameters, such as the memory overflow problem of the given data structure.

As a special Birth-Death process, the M/M/1 queueing model [57] can be formally analyzed by applying the verified properties in the last section of this chapter. Furthermore, a number of queueing systems can be expressed by means of instantiating the parameters of the verified Birth-Death process in this thesis, such as M/M/m (the m-server case) and M/M/∞ (Responsive Servers) [57], etc.

The applications presented Chapter 3 and 4 show that the formalizations of DTMCs and classified DTMCs are quite useful in the analysis of Markovian models using theorem proving technique. In fact, these formalizations facilitate diverse new research directions in the domain of formal verification, such as the formal analysis of hidden Markov models (HMMs) in higher-order logic, which can be applied in the recognition of DNA sequences. We will describe the formalization of HMMs and the verification of their important properties, as well as a DNA sequence analysis in HOL in the next chapter.

Chapter 5

Formalization of Hidden Markov Model

In this chapter, we provide the formalization of an extended DTMC models, namely, hidden Markov models (HMMs), which are the core concept for formally evaluating the probability of the occurrence of a particular observed sequence and finding the best state sequence to generate given observation. In order to present the usefulness of the formalization of HMM and the formal verification of HMM properties, we illustrate the formal analysis of a DNA (DeoxyribonNucleic acid) sequence at the end of the chapter.

5.1 Definition of HMM

In order to accurately analyze the HMMs, we propose to apply the formalized DTMC to formally define HMMs and verify their properties in higher-order logic as the extended DTMC models.

An HMM is a pair of two stochastic processes $\{X_k, Y_k\}_{k\geq 0}$, where $\{X_k\}_{k\geq 0}$ is a

Markov chain, and $\{Y_k\}_{k\geq 0}$ is *conditionally independent* [114] of $\{X_k\}$, i.e., Y_k depends only on X_k and not on any X_t, such that $t \neq k$. The HMMs model situations where an experimenter sees some observers at every instant (mathematically represented by Y_k) and suspects these observables to be the outcome of a process that can be modeled by a Markov chain ($\{X_k\}_{k\geq 0}$). The name "*Hidden* Markov Model" arises from the fact that the state in which this model is at a particular instant is not available to the observer. Now, a HMM is defined as a parameterized triple (A, B, $\pi(0)$) with the following conditions:

1. Hidden Markov Chain $\{X_k\}_{k\geq 0}$ with a finite state space S, the initial distribution $\pi(0) = \{\pi_i(0)\}_{i\in S}$ and the transition probabilities $A = \{a_{ij}\}_{i\in S, j\in S}$.

2. A random process $\{Y_k\}_{k\geq 0}$ with finite state space O. The hidden Markov chain and the random process are associated with the emission probabilities $B = \{b_j(O_k)\}_{j\in S, k\in O} = \{\mathcal{P}r\{Y_n = O_k | X_n = j\}\}_{j\in S, k\in O}$. It implies that:

   - $\forall\ j\ k.\ b_j(O_k) \geq 0$
   - $\sum_{k\in O} b_j(O_k) = 1$.

3. The random process $\{Y_k\}_{k\geq 0}$ and hidden Markov chain $\{X_k\}_{k\geq 0}$ have conditional independence.

   This yields the following formalization:

Definition 5.1. *(HMM)*

```
⊢ ∀ X Y p s_X s_Y p_0 p_ij p_XY.
  hmm X Y p s_X s_Y p_0 p_ij p_XY =
  dtmc X p s_X p_0 p_ij ∧ (∀ t.  random_variable (Y t) p s_Y) ∧
  (∀ i.  i ∈ space s_Y ⇒ {i} ∈ subsets s_Y) ∧
```

$$
\begin{array}{l}
(\forall\ \mathtt{t\ a\ i.}\ \ \mathbb{P}\{\mathtt{x\ |\ X\ t\ x = i}\} \neq \mathtt{0} \Rightarrow \\
\quad \mathbb{P}(\{\mathtt{x\ |\ Y\ t\ x = a}\}|\{\mathtt{x\ |\ X\ t\ x = i}\}) = \mathtt{p_{XY}\ t\ a\ i}) \wedge \\
\forall\ \mathtt{t\ a\ i}\ \mathtt{t}_{x_0}\ \mathtt{t}_{y_0}\ \mathtt{sts_X}\ \mathtt{sts_Y}\ \mathtt{ts_X}\ \mathtt{ts_Y}. \\
\quad \mathtt{t} \notin \{\mathtt{t}_{x_0} + \mathtt{m} \mid \mathtt{m} \in \mathtt{ts_X}\} \wedge \mathtt{t} \notin \{\mathtt{t}_{y_0} + \mathtt{m} \mid \mathtt{m} \in \mathtt{ts_Y}\} \wedge \\
\quad \mathbb{P}(\{\mathtt{x\ |\ X\ t\ x = i}\} \cap \bigcap_{k \epsilon ts_X} \{\mathtt{x\ |\ X\ (t}_{x_0} + \mathtt{k)\ x = EL\ k\ sts_X}\} \cap \\
\qquad \bigcap_{k \epsilon ts_Y} \{\mathtt{x\ |\ Y\ (t}_{y_0} + \mathtt{k)\ x = EL\ k\ sts_Y}\}) \neq \mathtt{0} \Rightarrow \\
\quad \mathbb{P}(\{\mathtt{x\ |\ Y\ t\ x = a}\}| \\
\qquad (\{\mathtt{x\ |\ X\ t\ x = i}\} \cap \bigcap_{k \epsilon ts_X} \{\mathtt{x\ |\ X\ (t}_{x_0} + \mathtt{k)\ x = EL\ k\ sts_X}\} \cap \\
\qquad \bigcap_{k \epsilon ts_Y} \{\mathtt{x\ |\ Y\ (t}_{y_0} + \mathtt{k)\ x = EL\ k\ sts_Y}\}))= \\
\quad \mathbb{P}(\{\mathtt{x\ |\ Y\ t\ x = a}\}|\{\mathtt{x\ |\ X\ t\ x = i}\})
\end{array}
$$

In this definition, the variable `X` denotes the random variable of the underlying DTMC (as the first conjunct constrains), `Y` indicates the random observations (so `Y t` is a random process as the second condition describes), and $\mathtt{p_{XY}}$ indicates the emission probabilities, i.e., the probability of obtaining a particular value for `Y` depending on the state `X`. Like the second condition in Definition 3.3, the condition $(\forall\ \mathtt{i.}\ \ \mathtt{i} \in \mathtt{space\ s_Y} \Rightarrow \{\mathtt{i}\} \in \mathtt{subsets\ s_Y})$ ensures that the event space is a discrete space. The conjunct $(\forall\ \mathtt{t\ a\ i.}\ \ \mathbb{P}\{\mathtt{x\ |\ X\ t\ x = i}\} \neq \mathtt{0} \Rightarrow \mathbb{P}(\{\mathtt{x\ |\ Y\ t\ x = a}\}|\{\mathtt{x\ |\ X\ t\ x = i}\}) = \mathtt{p_{XY}\ t\ a\ i})$ assigns the function $\mathtt{p_{XY}}$ to emission probabilities under the condition $\mathbb{P}\{\mathtt{x\ |\ X\ t\ x = i}\} \neq \mathtt{0}$, which ensures that the corresponding conditional probabilities are well-defined. The non-trivial conjunct in the above definition is the last one which formalizes the notion of conditional independence mentioned above. In our work, we consider mainly discrete time and finite-state space HMMs, which is the most frequently used case.

Time-homogenous HMMs can also be formalized in a way similar to time-homogenous DTMCs. Note that, in practice, time-homogenous HMMs always have a

finite state-space.

Definition 5.2. *(Time-homogeneous HMM)*

```
⊢ ∀ X Y p s_X s_Y p_0 p_ij p_XY.
    thmm X Y p s_X s_Y p_0 p_ij p_XY =
    hmm X Y p s_X s_Y p_0 p_ij p_XY ∧
    FINITE (space s_X) ∧ FINITE (space s_Y) ∧
    ∀ t a i j.  ℙ{x | X t x = i} ≠ 0 ∧ ℙ{x | X (t + 1) x = i} ≠ 0 ⇒
       (Trans X p s (t + 1) 1 i j = Trans X p s t 1 i j) ∧
       (p_xy (t + 1) i j = p_xy t i j)
```

where the model is constrained to be a hidden Markov model (by the first condition `hmm X Y p s_X s_Y p_0 p_ij p_XY`) with finite spaces for both the states and observations; also the last conjunct ensures that the transition probabilities of HMM satisfy the homogeneous property and the emission probabilities possess the independency of time.

Next, we verify some classical properties of HMMs, which play a vital role in reducing the user interaction for the formal analysis of systems that can be represented in terms of HMMs.

5.2 HMM Properties

HMMs are used to solve three types of problems: 1) evaluating the probability of occurrence of a particular observed sequence; 2) finding the most probable state sequence to generate given observations; and 3) learning parameters in the presumed model. The solutions are related to certain important HMM properties, which are verified in the following sections.

5.2.1 Joint Probability of HMM

The most important property of time-homogeneous HMMs is the expression of the joint distribution of a sequence of states and its corresponding observation, which can be expressed using products of its emission probabilities and transition probabilities. This is frequently used to find the best state path or estimate model's parameters. Mathematically, this is expressed as the following equation:

$$\mathcal{P}r(Y_0, \cdots, Y_t, X_0, \cdots, X_t) = \mathcal{P}r(X_0)\mathcal{P}r(Y_0|X_0)\prod_{k=0}^{t-1}\mathcal{P}r(X_{k+1}|X_k)\mathcal{P}r(Y_{k+1}|X_{k+1}) \tag{5.1}$$

and has been formally verified using the HOL theorem prover as follows:

Theorem 5.1. *(Joint Probability of HMM)*

```
⊢ ∀ X Y p sX sY p0 pij pXY t stsX stsY.
   thmm X Y p sX sY p0 pij pXY ⇒
   (ℙ(⋂_{k=0}^{t}{x | X k x = EL k stsX} ∩ ⋂_{k=0}^{t}{x | Y k x = EL k stsY}) =
    ℙ{x | X 0 x = EL 0 stsX}
    ℙ({x | Y 0 x = EL 0 stsY}|{x | X 0 x = EL 0 stsX})
    (PROD (0, t) (λ k.  ℙ({x | X (k + 1) x = EL (k + 1) stsX}|
                          {x | X k x = EL k stsX})
                        ℙ({x | Y (k + 1) x = EL (k + 1) stsY}|
                          {x | X (k + 1) x = EL (k + 1) stsX}))))
```

where the first eight variables keep the same notations as the corresponding ones in Definitions 5.1 and 5.2, variable `t` represents the index of the last observation considered in this theorem and it also equals to the amount of the production numbers from `0` to (`t - 1`) on the right-side of Equation (5.1), and $\texttt{sts}_\texttt{X}$ and $\texttt{sts}_\texttt{Y}$ denote the possible underlying state path and observable sequence, respectively.

Proof. The proof of this theorem is primarily based on Theorem 3.1 and Definitions 5.1 and 5.2, along with some arithmetic reasoning.

This theorem provides the foundations to solve the three types of problems that HMMs are primarily used for, as explained in Section 5.1.

5.2.2 Observation Sequence Probability

The first type of problems that HMMs are usually used to solve is evaluating the probability of occurrence of a particular observed sequence, which can be mathematically expressed as:

$$\mathcal{P}r\{Y_0, \cdots, Y_t\} = \sum_{\substack{X_0,\cdots,X_t \in \\ space\ s}} \mathcal{P}r\{X_0\}\mathcal{P}r\{Y_0|X_0\}\prod_{k=0}^{t-1}\mathcal{P}r\{X_{k+1}|X_k\}\mathcal{P}r\{Y_{k+1}|X_{k+1}\}$$

Using Theorem 3.1, we can formally verify this equation as follows.

Theorem 5.2. *(Joint Probability of Observation Sequence)*

```
⊢ ∀ X Y p s t sX sY p0 pij pXY stsY.
  thmm X Y p sX sY p0 pij pXY ⇒
    let ℒ = {L | EVERY (λx. x ∈ space sX) L ∧ (|L| = t + 1)} in
  (ℙ(⋂_{k=0}^{t} {x | Y k x = EL k stsY}) =
  SIGMA (λ stsX.  ℙ{x | X 0 x = EL 0 stsX}
                  ℙ({x | Y 0 x = EL 0 stsY}|{x | X 0 x = EL 0 stsX})
                  (PROD (0, t) (λ k.
                        ℙ({x | X (k + 1) x = EL (k + 1) stsX}
                          {x | X k x = EL k stsX})
                        ℙ({x | Y (k + 1) x = EL (k + 1) stsY}|
                          {x | X (k + 1) x = EL (k + 1) stsX}))) (stsX ∈ ℒ)
```

where |L| returns the length of the list L and EVERY p L is a predicate which is true iff the predicate p holds for every element of the list L.

Proof. The proof of this theorem is based on induction on variable t. The base case is proved by using some conditional probability theorems and set theoretic reasoning. The step case is then verified by applying the total probability theorem given in Equation (2.3c) and Theorem 5.1, as well as Definition 5.1.

This theorem is frequently used in estimating the parameters of a HMM using the maximum likelihood method [96], in which the computations of the parameters mainly depend on the joint probability of a given observable sequence. Furthermore, it plays an important role in the Baum-Welch algorithm [96].

5.2.3 Best Path Selection

In addition to the above property, researchers are often interested in the probability of a particular underlying state path, considering all possible observable sequence. The mathematical expression and the corresponding theorem are presented below.

$$\mathcal{P}r\{X_0, \cdots, X_t\} = \sum_{\substack{Y_0, \cdots, Y_t \in \\ space\ s1}} \mathcal{P}r\{X_0\}\mathcal{P}r\{Y_0|X_0\} \prod_{k=0}^{t-1} \mathcal{P}r\{X_{k+1}|X_k\}\mathcal{P}r\{Y_{k+1}|X_{k+1}\}$$

Theorem 5.3. *(Joint Probability of State Path)*

```
⊢ ∀ X Y p s t sX sY p0 pij pXY stsX.
  thmm X Y p sX sY p0 pij pXY ⇒
    let ℒ = {L | EVERY (λy. y ∈ space sY) L ∧ (|L| = t + 1)} in
  (ℙ(⋂_{k=0}^{t} {x | X k x = EL k stsX}) =
  SIGMA (λ stsY.  ℙ{x | X 0 x = EL 0 stsX}
                  ℙ({x | Y 0 x = EL 0 stsY}|{x | X 0 x = EL 0 stsX})
                  (PROD (0, t) (λ k.
```

$$\mathbb{P}(\{\texttt{x | X (k + 1) x = EL (k + 1) sts}_{\texttt{X}}\}$$
$$\{\texttt{x | X k x = EL k sts}_{\texttt{X}}\})$$
$$\mathbb{P}(\{\texttt{x | Y (k + 1) x = EL (k + 1) sts}_{\texttt{Y}}\}|$$
$$\{\texttt{x | X (k + 1) x = EL (k + 1) sts}_{\texttt{X}}\}))) \ (\texttt{sts}_{\texttt{Y}} \in \mathcal{L})$$

This theorem is very similar to Theorem 5.2 given the symmetric nature of the conditional independency property between the processes $\{X_k\}_{k\geq 0}$ and $\{Y_k\}_{k\geq 0}$. Hence, the proof process is also quite similar to that of Theorem 5.2.

Theorems 5.2 and 5.3 provide ways to *compute* the probabilities that are usually desired while analyzing HMMs, specifically, Theorem 5.3 is quite important in selecting the most probable state path (called *best path* in this thesis). Consequently, if the best path is to be selected among a series provided potential state pathes, then the joint probabilities of these state pathes can be calculated by instantiating the parameters with concrete values and a real number can be obtained for the corresponding state path, finally, the state path possessing the greatest joint probability will be selected.

5.3 Proof Automation

Though the analysis of HMMs is based on interactive theorem proving, it seems natural to try to *automatize* such computations. This is extremely useful since, in practice as one is always interested in applying the theorems to concrete situations. In this section, we describe how to automatically acquire interesting probabilities and find the best state path, for a given HMM, using the results of Theorems 5.2 and 5.3. This makes the accuracy of theorem proving available even to users with no knowledge about logic or theorem proving.

In order to automate the computation associated with Theorem 5.1, we define

a Standard ML (SML) function `hmm_joint_distribution` (`ini_distr trans_distr e_distr sts obs`) which takes as input the initial distributions, the transition probabilities, the emission distributions, a list of states and a list of observations: When calling this function, these parameters will be automatically substituted to, respectively, p_0, p_{ij}, p_{XY}, sts_X and sts_Y of Theorem 5.1. We then take `t` to be the length of `sts` (which should be the same as `obs`): this seems to be the most common case in practice, but could be easily relaxed if needed by adding a parameter to the function. We can then compute, using the theorems about list, real numbers, etc. in HOL4, the right-hand side of the equation in Theorem 5.1 in an exact way (as a fraction). In the end, the function returns the corresponding instantiation of a HOL4 theorem stating the equality between the joint probability and its value. Note that the result is really a HOL4 theorem: even the operations between real numbers like multiplication or addition are obtained by deductive reasoning, thus making every single step of the computation completely reliable and *traceable*. For convenience, the result can also be converted (outside HOL4) into an SML floating point value, in order to compare with those results created by simulation tools.

The implementation of the function `hmm_joint_distribution` requires the development of an intermediate lemma, in which some functions are defined for parameterizing the variables in Theorem 5.1 and outputting the results given by HOL4 through SML.

The computations associated with Theorem 5.3 can also be automated similarly. To obtain the best path automatically, we need to compute the set of all possible state paths, compute the probability of each of these paths as the function `hmm_joint_distribution` does, and then return the path which has the highest joint

probability. In order to be the most accurate as possible, all these computations shall be done inside HOL4. This can be achieved by an SML function `best_path` (`ini_distr trans_distr e_distr st_ty obs`) where `ini_distr`, `trans_distr`, `e_distr`, and `obs` denote the same objects as for `hmm_joint_distribution` and `st_ty` denotes the type of terms representing states. This type should be a non-recursive enumerated type, i.e., defined as $\mathtt{C_1} \mid \mathtt{C_2} \mid \ \ldots\ \mathtt{C_k}$, where $\mathtt{C_1}, \ldots, \mathtt{C_k}$ are constructors without arguments: this ensures that the state-space is finite. The function then takes care of computing the list of all possible paths, then computes the corresponding joint probability as `hmm_joint_distribution` does, and, in the end, returns the state path which has the best such probability (note that the notion of "best probability" is also defined inside HOL4 by using the axiomatic definition of the order on real numbers).

This function is currently very slow due to the computation of the set of all possible state paths, but there is a lot of room for improvement, in particular by filtering paths which have trivially a null probability. This can be done by proving a theorem which is quite similar as Theorem 5.3 but the set of the possible state path does not include those containing null transition probability or null emission probability, e.g.,

$$
\begin{aligned}
\mathcal{L} = \{\mathtt{L} \mid\ & \mathtt{EVERY}\ (\lambda\ \mathtt{x}.\ \ \mathtt{x} \in \mathtt{space}\ \mathtt{s_X})\ \mathtt{L} \wedge (|\mathtt{L}| = \mathtt{n} + 1) \wedge \\
& (\forall\ \mathtt{x}\ \mathtt{y}\ \mathtt{k}.\ \ \mathtt{x} \in \mathtt{space}\ \mathtt{s_X} \wedge \mathtt{y} \in \mathtt{space}\ \mathtt{s_Y} \Rightarrow \\
& \qquad \mathbb{P}(\{\mathtt{x} \mid \mathtt{X}\ (\mathtt{k}+1)\ \mathtt{x} = \mathtt{EL}\ (\mathtt{k}+1)\ \mathtt{sts_X}\} \mid \\
& \qquad\quad \{\mathtt{x} \mid \mathtt{X}\ \mathtt{k}\ \mathtt{x} = \mathtt{EL}\ \mathtt{k}\ \mathtt{sts_X}\}) > 0 \wedge \\
& \qquad \mathbb{P}(\{\mathtt{x} \mid \mathtt{Y}\ \mathtt{k}\ \mathtt{x} = \mathtt{EL}\ \mathtt{k}\ \mathtt{sts_X}\} \mid \\
& \qquad\quad \{\mathtt{x} \mid \mathtt{X}\ \mathtt{k}\ \mathtt{x} = \mathtt{EL}\ \mathtt{k}\ \mathtt{sts_X}\}) > 0)\} \ .
\end{aligned}
$$

For those applications containing any null transition probabilities or emission probabilities, the computation load will be significantly reduced by applying a theorem which is similar to Theorem 5.3, except for the L.

We now show how to apply these theorems and functions in practice, by providing the formal analysis of a HMM of DNA model in the next section.

5.4 Application: DNA Sequence Analysis

DNA sequence analysis plays a vital role in constructing gene mapping, discovering new species and investigating disease-manifestations in genetic linkage, parental testing and criminal investigation. Statistical methods are mainly applied for analyzing DNA sequences. In particular, obtaining the probability of a state path underlying the DNA fragment is the most critical step in identifying a particular DNA sequence.

A DNA fragment is a sequence of proteins called A, T, G and C. However, not every sequence represents a valid DNA: some regularities can be found among the possible sequences. For instance, it might be that all four proteins can appear with equal probability at the beginning of the sequence, but, after a particular point, only A and G can appear, and then all four can appear again but with higher probabilities for A and E. In this application, there are thus three different "states" of the DNA, characterized by the probabilities of occurrence of each base. In this DNA model, the first state is called *exon* (E), the second one *5' splice site* (5), and the third one *intron* (I) [10]. This model is described and studied very naturally using HMMs [21]: a DTMC over the states E, 5, and I is used in order to know in which state the proteins are, then another random process is defined which characterizes the emission of A, G, T or C according to the state which the proteins are in. This is summarized in Figure 5.1.

In order to formalize this HMM, we first define the types representing the states and the bases below.

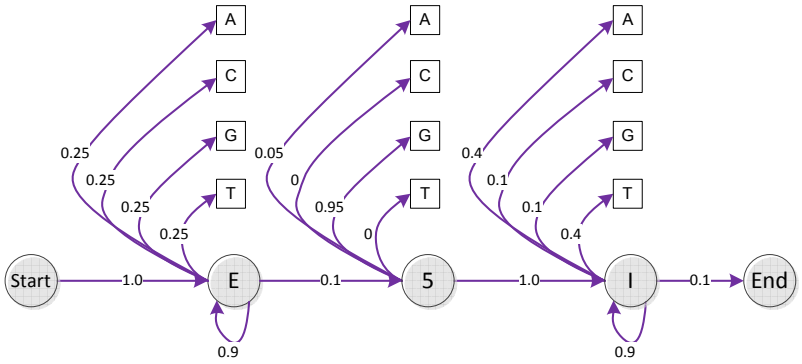

Figure 5.1: 5' Splice Site Recognition Model

Definition 5.3. *(Data Types)*

$\vdash$ `dna = A | G | T | C`

$\vdash$ `state = START | E | I | FIVE | END`

Note that, in order to characterize the sequence, it is a common practice to add some fake *start* and *end* states. Hence the definition of `state` in Definition 5.3 includes `START` and `END`, which have no emission probabilities. As examples, we define the following state and DNA sequences:

Definition 5.4. *(State Path and DNA Sequences)*

$\vdash$ `state_seq` $=$

$$[\texttt{START}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{E}; \texttt{FIVE}; \texttt{I}; \texttt{I}; \texttt{I}; \texttt{I}; \texttt{I}; \texttt{I}; \texttt{I}; \texttt{END}]$$

$\vdash$ `dna_seq` $= [\texttt{C}; \texttt{T}; \texttt{T}; \texttt{C}; \texttt{A}; \texttt{T}; \texttt{G}; \texttt{T}; \texttt{G}; \texttt{A}; \texttt{A}; \texttt{A}; \texttt{G}; \texttt{C}; \texttt{A}; \texttt{G}; \texttt{A}; \texttt{C}; \texttt{G}; \texttt{T}; \texttt{A}; \texttt{A}; \texttt{G}; \texttt{T}; \texttt{C}; \texttt{A}]$

So to model the HMM represented in Figure 5.1, we need an initial distribution, the transition probabilities, and the emission probabilities, which we define as follows:

Definition 5.5. *(DNA Model Parameters)*

$\vdash$ `ini_distr i` $=$ `if (i = START) then 1 else 0`

$\vdash$ trans_distr t i j = case (i, j) of

(START, E) $\rightarrow$ 1 | (E, E) $\rightarrow$ 0.9 | (E, FIVE) $\rightarrow$ 0.1 | (FIVE, I) $\rightarrow$ 1 |

(I, I) $\rightarrow$ 0.9 | (I, END) $\rightarrow$ 0.1 | _ $\rightarrow$ 0

$\vdash$ e_distr a i = case (i, a) of

(E, _) $\rightarrow$ 0.25 | (FIVE, A) $\rightarrow$ 0.05 | (FIVE, G) $\rightarrow$ 0.95 | (I, A) $\rightarrow$ 0.4 |

(I, T) $\rightarrow$ 0.4 | (I, C) $\rightarrow$ 0.1 | (I, G) $\rightarrow$ 0.1 | _ $\rightarrow$ 0

Then, in order to work with random variables X and Y denoting the states and the observations, respectively, on a probability space p, it is sufficient to have the following predicate:

thmm X Y p s_X s_Y ini_distr trans_distr e_distr

$\wedge$ space s_X = univ(: state) $\wedge$ space s_Y = univ(: dna)

where univ(:t) is the set of all possible values of type t, e.g., univ(:dna) = {A; G; T; C}. Now, for instance, we can prove the theorem which gives the probability of obtaining the sequence dna_seq if the underlying state path is state_seq:

Theorem 5.4. *(Joint Probability of A DNA Segment)*

$\vdash \forall$ X Y p s_X s_Y.

thmm X Y p s_X s_Y ini_distr trans_distr e_distr $\wedge$

space s_X = univ(: state) $\wedge$ space s_Y = univ(: dna) $\Rightarrow$

$\mathbb{P}(\bigcap_{k=0}^{|\text{state\_seq}|-1}\{\text{x} \mid \text{X k x = EL k state\_seq}\} \bigcap$

$\bigcap_{k=0}^{|\text{dna\_seq}|-1}\{\text{x} \mid \text{Y k x = EL k dna\_seq}\}) =$

$0.25^{18} * 0.9^{23} * 0.1^4 * 0.95 * 0.4^5$

To prove this theorem, a lemma of Theorem 5.1 is first verified:

Lemma 5.1. *(Extended Joint Probability of HMM)*

$\vdash \forall$ X Y p t s_X s_Y p_0 p_{ij} p_{XY} sts_X sts_Y.

```
thmm X Y p s_X s_Y p_0 p_ij p_XY ∧ (|sts_x| = t + 3) ∧ (|sts_y| = t + 1) ⇒
(ℙ(⋂_{k=0}^{t+2}{x | X k x = EL k sts_X} ⋂ ⋂_{k=0}^{t}{x | Y k x = EL k sts_Y}) =
ℙ{x | X 0 x = EL 0 sts_X}
ℙ({x | X (k + 2) x = EL (k + 2) sts_X}|
  {x | X (k + 1) x = EL (k + 1) sts_X})
(PROD (0, t) (λ k.  ℙ({x | X (k + 1) x = EL (k + 1) sts_X}|
                      {x | X k x = EL k sts_X})
                    ℙ({x | Y (k + 1) x = EL k sts_Y}|
                      {x | X k x = EL (k + 1) sts_X}))))
```

where the state path $\texttt{sts}_\texttt{x}$ involves the START and END states, as shown in Figure 5.1. Lemma 5.1 allows us to consider the joint probability of the states along with the observed events, in which the number of states is more than the observations, in a HMM, comparing to the Theorem 5.1.

Another interesting property is to find the state path has the best probability of occurrence given a particular DNA sequence. In our particular context, this problem is called *5' splice site recognition*. We can analyze the DNA segment, which starts from any potential state. This can be formalized as follows using the previously used DNA sequence:

Theorem 5.5. *(Best State Path)*

```
⊢ ∀ X Y p s_X s_Y.
  thmm X Y p s_X s_Y ini_distr trans_distr e_distr ∧
  space s_X = univ(: state)  ∧  space s_Y = univ(: dna)  ⇒
  REAL_MAXIMIZE_SET
        [E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; FIVE; I; I; I; I; I; I; I]
        (λsts. ℙ(⋂_{k=0}^{|sts|−1}{x | X k x = EL k state_seq} ⋂
```

$$\bigcap_{k=0}^{|\texttt{dna\_seq}|-1}\{\texttt{x} \mid \texttt{Y k x} = \texttt{EL k dna\_seq}\})$$

$$\{\texttt{sts} \mid |\texttt{sts}| = 26\})$$

where `REAL_MAXIMIZE_SET m f s` is a predicate which is true only if `f m` is the maximum element of $\{\texttt{f x} \mid \texttt{x} \in \texttt{s}\}$ (this is defined as a predicate because there can be several elements of `s` having this property). Note once again that this theorem is formally verified, i.e., even the comparisons between probabilities are proved deductively from the axiomatic definition of real numbers. Consequently, the confidence that we can have in the result is maximal.

While Theorems 5.4 and 5.5 have been proved in the classical theorem proving way, i.e., interactively, there are rare chances that a biologist has the required knowledge of higher-order logic and HOL4 so as to conduct such a study. However, we automate the analysis by using SML functions that we presented in the previous section.

5.5 Summary and Discussion

In this chapter, we first provided a formal definition of hidden Markov models. Building upon the definition of the time-homogeneous HMM, we verified fundamental properties, such as the *joint probability of the sequences of states and observations*, the *joint probability of an observed sequence* and *the best path selection*. These properties provide the foundations of the computation algorithms applied in diverse simulation tools and computer algebra systems in order to mitigate the tremendous computation loads in the HMM analysis. The HOL4 proof script of above theorems consists of about 1600 lines of code. In addition, we presented an automatic verification method for systems involving HMMs in this chapter. Our automation can be further optimized in order to compare with the other simulation tools.

HMMs are widely applied in almost all speech recognition [114], data compression and artificial intelligence and pattern recognition, as well as computational molecular biology applications. For this reason, we utilized our proved results for the formal analysis of a DNA sequence. We first defined the DNA types and the underlying states using HOL types, then constructed the DNA sequence model by instantiating the relevant parameters. The joint probability of a state sequence is obtained accurately based on the given model. Moreover, we described how to select the DNA sequence, which has the highest joint probability of a possible underlying state path (the best path). Finally, we showed the way to automatically compute the best path of an observed DNA sequence.

This HMM application is a vivid example showing that the formalized DTMC (presented in Chapter 3) can be easily applied to formalize various other derivative Markovian processes, such as Queue, semi-Markov processes, and Markov Decision Process (MDP) as well. Numerous algorithms implemented in probabilistic model checker for analyzing DTMC models can be verified using theorem proving [47]. Similarly, the algorithms, such as *Forward-Backward*, *Viterbi* and *Baum-Welch* algorithm [74], implemented in many statistical simulation tools for analyzing discrete HMMs can also be verified by using the verified HMMs theorems interpreted in Chapter 5. These algorithms can then be integrated into a tool to formally analyze HMMs automatically by using automated theorem proving techniques.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we have proposed a framework (shown in Figure 1.2) to facilitate the formal analysis of the systems modeled as discrete-time Markov chains using higher-order logic theorem proving. We formalized *Discrete-Time Markov Chain*, *Classified States* and *Classified DTMCs* as well as *Hidden Markov Model* and verified their most important properties, which are the major components in our framework, in the theorem prover HOL4. These formalizations offer the capability of formally evaluate the performance, maintainability and reliability of diverse systems which are described as DTMCs. Compared with conventional paper-and-pencil analysis, simulation technique or computer algebra systems, our approach allows the formal verification of the desired DTMC systems using a sound theorem prover and thus guarantees generic, accurate and reliable results. Thus, we believe that the analysis of DTMCs using higher-order logic theorem proving based on our development will be free of approximation and precision problems due to the soundness nature of higher-order logic

environment. For these reasons, the proposed approach can be used in formal performance analysis of safety critical and/or mission critical engineering and scientific applications related to discrete-time Markov chain.

Among the higher-order logic libraries shown in the framework, the generalized formalization of discrete-time Markov chain is built upon a state-of-the-art probability theory and allows to handle both time-homogeneous and time-inhomogeneous DTMCs with generic state spaces. Based on this formalization, we have been able to formally verify a fundamental telecommunication network element, the binary communication channel as well as the tag-collection process in an automatic mail quality measurement (AMQM) protocol, which are expressed as discrete-time Markov chains. These applications highlight the benefits of the formalization of DTMCs and the formal verification of their properties using a higher-order-logic theorem prover.

The formalization of the DTMC theory enabled the formal definitions of classified states and classified DTMCs and the formal proofs of their significant properties. As an example, we used the formalization of the aperiodic and irreducible DTMCs to validate the LRU stack model in higher-order logic and achieved a general result which is consistent with a simulation based analysis of a similar model [106]. In addition, we presented a formalization of discrete-time Birth-Death process, which can be applied in analyzing software reliability, based on the formalization of classified DTMCs. These examples highlight the benefits of our results and guarantee the validity of every single instance of the system.

Finally, we used the formalized DTMC to model Hidden Markov Chains (HMMs), which is a widely used type of extended DTMC with diverse potential applications. The proposed formalization of HMMs provides a novel approach to analyze statistical models involving two random processes. Also, we presented a case study about DNA

sequence analysis based on a HMM. To facilitate this analysis process in a better way, we introduced certain automatic simplifiers to reduce the user intervention in formal modeling and analyzing of real-world systems that can be described in terms of HMMs.

6.2 Future Work

The Markov Chain theory involves important mathematical concepts in analyzing a variety of applications and the DTMC concept is one of its basic brick. The formalization and verification results, presented in this thesis, pave the avenues to a precise analysis of Markovian systems using theorem proving as a complement to the traditional paper-and-pencil, simulation and computer algebra system based analyses, as well as probabilistic model checking techniques. Diverse future work directions can be performed building upon the work presented in this thesis. These potential directions are depicted in the Figure 6.1, and discussed briefly the sequel.

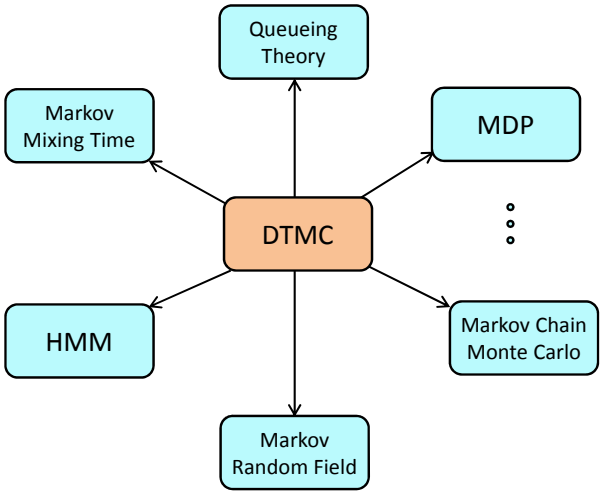

Figure 6.1: Future Research Directions

- A continuous-time Markov chain (CTMC) refers to a random process, in which the random variables remain in the current state for some random (particularly, exponentially distributed) interval of time and then transit to different states. Using the formal definition of the continuous-time Markov chain and two of its verified properties in [70] and the limiting, derivative and integral theories, it is possible formally derive the *Kolmogorov's backward equation* [109] as well as variety of its other properties in HOL4. The formalization of CTMC would enable the formal analysis of many applications, e.g., continuous-time HMMs, which are used for formally assessing diseases (i.e. the progress of breast cancer) in medical and biological domains, or estimating the electronic system reliability. Continuous-time semi-Markov processes [51] and Markov jump processes [78] can also be formalized using such a CTMC formalization.

- Queueing theory [11] is a mathematical study of a system which processes of the flows of customers or/and services. A queue system is usually described as a Birth-Death process in which the population consists of customers. This potential project may start by formalizing the *count process* and *poisson process* [112], as well as the continuous-time Markov chain. Various queues can be found in the open literature, such as M/G/1, M/G/∞, GI/GI/1 [101] and they can be used to model diverse interesting applications.

- The principal of Markov Decision Process (MDP) [52] is a Markov chain with a reward function and a discount factor. We can develop a platform for formally analyzing MDP models using our formalization of DTMC. For instance, the formalization of discrete-time MDP is just a simple extension of Definition 3.1. It can also be further extended to formalize continuous-time MDP [31], which can

be applied in the analysis of *queueing systems*, *epidemic processes* and *population processes* (including *Birth-Death Processes*, *Birth*, *Death* and *Catastrophe Processes*) [27].

- *Markov Random Field* (MRF) (also called *undirected graphical model*)[56] is a type of stochastic process, which forms a natural generalization of Markov processes, where the time index is replaced by a space index. The MRF is mainly found in the physics domains and the basic ideas of this subject and its applications can be found in [56]. Our formalization of DTMC provides a general form for formally defining a MRF from the discrete space point of view.

- The random walk is a mathematical model describing a path that consists of a succession of random steps. In fact, the random walk process is a *sum process* [66] and this type of random process has several different classes, such as the *symmetric random walk* and the *asymmetric simple random walk* [59]. Among the diverse random walk processes, the *simple random walk* (or *nearest neighbor random walk*) in one dimension is a DTMC. The formalization of random walk can start from the simple random walk and then be extended to other types of random walk processes. The applications of random walk can be found in diverse areas, such as polymer physics, kinetic theory of chemical reactions [59].

- The Gambler's ruin problem is a simple random walk, however, many variations of this problem can be found in the open literature, such as the *fair ruin problem*, the *unfair ruin problem* and the *N-player ruin problem*. The major issue of this problem is to find out the solution, which refers to the probability that the gambler wins finally. This can be formally proved in a theorem prover using higher-order logic by applying our formalization of DTMC interpreted in this

thesis. The principal of the gambler's ruin problem offers the capabilities of modeling the *risk insurance business* [38] in higher-order logic.

- Our work provides a possibility to combine both model checking and theorem proving in order to offer an integrated framework that takes advantage of both methods for formally analyzing Markovian systems. For instance, the behavior of a timed Markovian system involves huge computations in model checking that may lead to state space explosion problem. Since probabilistic model checkers, such as PRISM, are based on probabilistic extensions of the timed automata formalism, an interface between probabilistic PRISM model and HOL4 will only require the definition of the timed automaton in HOL4. This interface can utilize our proved theorems in HOL4 in order to verify such behavior. On the other hand, in order to improve the automation of the verification process in HOL4, certain HOL4 goals can be translated into PCTL expressions, and hence verified automatically using a probabilistic model checker.

- The formalizations involved in our work can also be applied in formal analysis of diverse real-world applications, such as telecommunication networks [76], real-time reactive systems [89], robotic system [26] and digital circuits [62] and so on.

Bibliography

[1] ISO/IEC 18000-7 Information Technology. RFID for Item Management Part 7: Parameters for Active Air Interface Communications at 433 MHz, 2008.

[2] R. Affeldt and M. Hagiwara. Formalization of Shannon's Theorems in SSReflect-Coq. In *Interactive Theorem Proving*, volume 7406 of *LNCS*, pages 233–249. Springer, 2012.

[3] R. B. Ash. *Basic Probability Theory*. John Wiley & Sons, 1970.

[4] B. Avi-Itzhak and D. P. Heyman. Approximate Queuing Models for Multiprogramming Computer Systems. *Operations Research*, 21(6):pp. 1212–1230, 1973.

[5] C. Baier and J. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[6] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, USA, 1994.

[7] Marco Bernardo and Roberto Gorrieri. Extended markovian process algebra. In *Concurrency Theory, 7th International Conference, Pisa, Italy, August 26-29, 1996, Proceedings*, volume 1119 of *LNCS*, pages 315–330. Springer, 1996.

[8] R. N. Bhattacharya and E. C. Waymire. *Stochastic Processes with Applications*. John Wiley & Sons, 1990.

[9] J. Bialas. The σ-Additive Measure Theory. *Journal of Formalized Mathematics*, 2, 1990.

[10] P. Bjesse. What is Formal Verification? *SIGDA E-newsletter*, 35(24):1, 2005.

[11] P. Brémaud. *Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues*. Springer, 1998.

[12] ChIP-Seq Tool Set. http://havoc.genomecenter.ucdavis.edu/cgi-bin/chipseq.cgi, 2013.

[13] K.L. Chung. *Markov chains with stationary transition probabilities*. Springer, 1960.

[14] G. Ciardo, J. K. Muppala, and K. S. Trivedi. SPNP: Stochastic Petri Net Package. In *Workshop on Petri Nets and Performance Models*, pages 142–151, 1989.

[15] A. Coble. *Anonymity, Information, and Machine-Assisted Proof*. Ph.D Thesis, University of Cambridge, UK, 2009.

[16] A. R. Coble. *Anonymity, Information, and Machine-Assisted Proof*. PhD thesis, University of Cambridge, Cambridge, UK, 2010.

[17] Coq. http://coq.inria.fr/, 2013.

[18] M. Tessmer D. H. Jonassen and W. H. Hannum. *Task Analysis Methods for Instructional Design*. Lawrence Erlbaum, 1999.

[19] D. M. Davis. Markov Analysis of APBA, a Baseball Simulation Game. *Journal of Quantitative Analysis in Sports*, 7(3), July, 2011.

[20] M. Duflot, M. Kwiatkowska, G. Norman, and D. Parker. A Formal Analysis of Bluetooth Device Discovery. *Journal on Software Tools for Technology Transfer*, 8(6):621–632, 2006.

[21] S. R. Eddy. What is a Hidden Markov Model? *Nature Biotechnology*, 22(10):1315–1316, 2004.

[22] A. Fehnker and P. Gao. Formal Verification and Simulation for Performance Analysis for Probabilistic Broadcast Protocols. In *Ad-Hoc, Mobile, and Wireless Networks*, volume 4104 of *LNCS*, pages 128–141. Springer, 2006.

[23] S. Frédéric and M. Delorenzi. MAMOT: Hidden Markov Modeling Tool. *Bioinformatics*, 24(11):1399 – 1400, 2008.

[24] J. Franklin. *Matrix theory*. Dover Publications, 2000.

[25] D. Gamerman and H. F. Lopes. *Fundamentals of Applied Probability Theory*. Chapman & Hall/CRC, 2006.

[26] E. Gjondrekaj, M. Loreti, R. Pugliese, F. Tiezzi, C. Pinciroli, M. Brambilla, M. Birattari, and M. Dorigo. Towards a Formal Verification Methodology for Collective Robotic Systems. In *14th International Conference on Formal Engineering Methods*, pages 54–70, 2012.

[27] W. Goffman. An Epidemic Process in an Open Population. *Nature*, 205(4973):831–832, February 1965.

[28] M.J.C. Gordon. Mechanizing Programming Logics in Higher-0rder Logic. In *Current Trends in Hardware Verification and Automated Theorem Proving*, pages 387–439. Springer, 1989.

[29] K. Gŏseva-Popstojanova and K. S. Trivedi. Failure Correlation in Software Reliability Models. *IEEE Transaction on Reliability*, 49:232, 2000.

[30] GreatSPN. http://www.di.unito.it/∼greatspn/index.html, 2013.

[31] X. Guo and O. Hernández-Lerma. *Continuous-Time Markov Decision Processes: Theory and Applications*. Stochastic Modelling and Applied Probability. Springer, 2009.

[32] P. J. Haas. *Stochastic Petri Nets: Modelling, Stability, Simulation*. Springer, 2002.

[33] O.J. Haggarty, W.J. Knottenbelt, and J.T. Bradley. Distributed Response Time Analysis of GSPN Models with MapReduce. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pages 82–90, 2008.

[34] O. Häggström. *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, 2002.

[35] J. D. Hamilton. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, (2):357–384.

[36] J. Harrison. *Theorem Proving with the Real Numbers*. Springer, 1998.

[37] J. Harrison and L. Théry. A Skeptic's Approach to Combining HOL and MAPLE. *Journal of Automated Reasoning*, 21:279–294, 1998.

[38] J. Michael Harrison. Ruin Problems with Compounding Assets. *Stochastic Processes and their Applications*, 5(1):67 – 79, 1977.

[39] O. Hasan. *Formal Probabilistic Analysis using Theorem Proving*. PhD Thesis, Concordia University, Montreal, QC, Canada, 2008.

[40] O. Hasan, N. Abbasi, B. Akbarpour, S. Tahar, and R. Akbarpour. Formal Reasoning about Expectation Properties for Continuous Random Variables. In *Formal Methods*, volume 5850 of *LNCS*, pages 435–450. Springer, 2009.

[41] O. Hasan and S. Tahar. Reasoning about Conditional Probabilities in a Higher-Order-Logic Theorem Prover. *Journal of Applied Logic*, 9(1):23 – 40, 2011.

[42] HMMER. http://hmmer.janelia.org/, 2013.

[43] HMMTool. http://iri.columbia.edu/climate/forecast/stochastictools/, 2013.

[44] HOL-Light. http://www.cl.cam.ac.uk/∼jrh13/hol-light/, 2013.

[45] HOL4. http://hol.sourceforge.net/, 2013.

[46] J. Hölzl and A. Heller. Three Chapters of Measure Theory in Isabelle/HOL. In *Interactive Theorem Proving*, volume 6898 of *LNCS*, pages 135–151. Springer, 2011.

[47] J. Hölzl and T. Nipkow. Verifying pCTL Model Checking. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 7214 of *LNCS*, pages 347–361. Springer, 2012.

[48] J. J. Hopfield. Neurocomputing: Foundations of Research. pages 457–464. MIT Press, 1988.

[49] J. Hurd. *Formal Verification of Probabilistic Algorithms*. PhD Thesis, University of Cambridge, UK, 2002.

[50] Isabellel. http://isabelle.in.tum.de/, 2013.

[51] J. Janssen and R. Manca. Applied Semi-Markov Processes. pages 481–515, 2010.

[52] B. Jeannet, P.D. Argenio, and K. Larsen. RAPTURE: A Tool for Verifying Markov Decision Processes. In *Conference on Concurrency Theory*, pages 84–98, Brno, Czech Republic, 2002.

[53] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, 2007.

[54] M. Kaliakatsos-Papakostas, M.G. Epitropakis, and M. N. Vrahatis. Weighted Markov Chain Model for Musical Composer Identification. In *Applications of Evolutionary Computation*, volume 6625 of *LNCS*, page 334–343, 2011.

[55] D. Kannan. *An Introduction to Stochastic Processes*. Elsevier North Holland, 1979.

[56] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. http://www.ams.org/books/conm/001/, AMS Book Online, 2013.

[57] L. Kleinrock. *Queueing Systems*, volume I: Theory. Wiley Interscience, 1975.

[58] A. N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, 1933. English translation (1950): Foundations of the Theory of Probability. Chelsea Publishing Co.

[59] T. Konstantopoulos. Introductory Lecture Notes on Markov Chains and Random Walks. 2009.

[60] Y. Kovchegov. A Note on Adiabatic Theorem for Markov Chains. *Statistics & Probability Letters*, 80(3-4):186–190, 2010.

[61] P. Kritzinger and F. Bause. *Introduction to Stochastic Petri Net Theroy*. Vieweg Verlag, 1995.

[62] J. A. Kumar. *Statistical Guarantees of Performance for RTL Designs*. PhD thesis, University of Illinois at Urbana-Champaign, USA, 2012.

[63] M. Kwiatkowska, G. Norman, and J. Sproston. Probabilistic Model Checking of the IEEE 802.11 Wireless Local Area Network Protocol. In *Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, volume 2399 of *LNCS*, pages 169–187. Springer, 2002.

[64] H.G. Landau. On Dominance Relations and the Structure of Animal Societies: II. Some Effects of Possible Social Factors. *The Bulletin of Mathematical Biophysics*, 13(4):245–262, 1951.

[65] V. Lecomte, C. Appert-Rolland, and F. van Wijland. Thermodynamic Formalism for Systems with Markov Dynamics. *Journal of Statistical Physics*, 127:51–106, 2007.

[66] A. Leon-Garcia. *Probability, Statistics, and Random Processes for Electrical Engineering*. Pearson Education, 2007.

[67] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2006.

[68] L. Liu. http://hvg.ece.concordia.ca/code/hol/cdtmc/, 2013.

[69] L. Liu, O. Hasan, and S. Tahar. Formalization of Finite-State Discrete-Time Markov Chains in HOL. In *Automated Technology for Verification and Analysis*, volume 6996 of *LNCS*, pages 90–104. Springer, 2011.

[70] L. Liu, O. Hasan, and S. Tahar. On the Formalization of Continuous-Time Markov Chains in HOL. Technical Report, Concordia University, April, 2013.

[71] L. Liu, O. Hasan, and S. Tahar. Formalization of Discrete-Time Markov Chains in HOL. Technical Report, Concordia University, December, 2010.

[72] I. L. MacDonald and W. Zucchini. *Hidden Markov and Other Models for Discrete-valued Time Series*. Chapman & Hall, London, 1997.

[73] J. Majewski, H. Li, and J. Ott. The Ising Model in Physics and Statistic Genetics. *The American Journal of Human Gentics*, 69(4):853 – 862, 2001.

[74] V. M. Mäntylä and V. T. Tutkimuskeskus. *Discrete Hidden Markov Models with Application to Isolated User-dependent Hand Gesture Recognition*. VTT publications. Technical Research Centre of Finland, 2001.

[75] Maple. http://www.maplesoft.com, 2013.

[76] M. Martal, S. Busanelli, and G. Ferrari. Markov Chain-based Performance Analysis of Multihop IEEE 802.15.4 Wireless Networks. *Performance Evaluation*, 66(12):722 – 741, 2009.

[77] Mathematica. www.wolfram.com, 2013.

[78] P. Metzner, E. Dittmer, T. Jahnke, and Ch Schütte. Generator Estimation of Markov Jump Processes. *Journal of Computational Physics*, 227(1):353–375, 2007.

[79] T. Mhamdi. *Information-Theoretic Analysis using Theorem Proving*. PhD Thesis, Concordia University, Montreal, QC, Canada, 2012.

[80] T. Mhamdi, O. Hasan, and S. Tahar. On the Formalization of the Lebesgue Integration Theory in HOL. In *Interactive Theorem Proving*, volume 6172 of *LNCS*, pages 387–402. Springer, 2010.

[81] T. Mhamdi, O. Hasan, and S. Tahar. Formalization of Entropy Measures in HOL. In *Interactive Theorem Proving*, volume 6898 of *LNCS*, pages 233–248. Springer, 2011.

[82] R. Milner. A Theory of Type Polymorphism in Programming. *Journal of Computer and System Sciences*, 17:348–375, 1977.

[83] Mizar. http://www.mizar.org/, 2013.

[84] Mobius. http://www.mobius.illinois.edu/, 2013.

[85] W. J. Knottenbelt N. J. Dingle and P. G. Harrison. HYDRA - Hypergraph-based Distributed Response-time Analyser. In *International Conference on Parallel and Distributed Processing Technique and Applications*, pages 215 – 219, 2003.

[86] A. Nedzusiak. σ-fields and Probability. *Journal of Formalized Mathematics*, 1:1–6, 1989.

[87] B. Nokovic and E. Sekerinski. Analysis of Interrogator-tag Communication Protocols. Technical Report, McMaster University, 2010.

[88] J. R. Norris. *Markov Chains*. Cambridge University Press, 1999.

[89] O. Ormandjieva, V. S. Alagar, and M. Zheng. Early Quality Monitoring in the Development of Real-time Reactive Systems. *Journal of Systems and Software*, 81(10):1738–1753, 2008.

[90] D. A. Parker. *Implementation of Symbolic Model Checking for Probabilitics Systems*. PhD Thesis, University of Birmingham, Birmingham, UK, 2002.

[91] L.C. Paulson. *Isabelle: A Generic Theroem Prover*, volume 828 of *LNCS*. Springer, 1994.

[92] PEPA. http://www.dcs.ed.ac.uk/pepa/, 2013.

[93] N. R. Prasad, R. C. Ender, S. T. Reilly, and G. Nesgos. Allocation of Resources on a Minimized Cost Basis. In *IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes*, volume 13, pages 402–403, 1974.

[94] PRISM. http://www.prismmodelchecker.org, 2013.

[95] PVS. http://pvs.csl.sri.com/, 2013.

[96] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., 1990.

[97] L. E. Reichl. *A Modern Course in Statistical Physics*, volume 71. University of Texas Press Austin, USA, 1980.

[98] A. W. Robertson, S. Kirshner, and P. Smyth. Downscaling of Daily Rainfall Occurrence over Northeast Brazil using a Hidden Markov Model. *Journal of Climate*, 17:4407–4424, November 2004.

[99] T. Rollins and W. D. Strecker. Use of the LRU Stack Depth Distribution for Simulation of Paging Behavior. *Communications of the ACM*, 20(11):795–798, 1977.

[100] J. Rutten, M. Kwaiatkowska, G. Normal, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems*, volume 23 of *CRM Monograph Series*. American Mathematical Society, 2004.

[101] T. L. Saaty. *Elements of Queueing Theory: with Applications*. McGraw-Hill, 1961.

[102] M. Sczittnick. MACOM - A Tool for Evaluating Communication Systems. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 7–10, 1994.

[103] K. Sen, M. Viswanathan, and G. Agha. VESTA: A Statistical Model-Checker and Analyzer for Probabilistic Systems. In *IEEE International Conference on the Quantitative Evaluation of Systems*, pages 251–252, 2005.

[104] L. Shapiro and D. Zeilberger. A Markov Chain Occurring in Enzyme Kinetics. *Journal of Mathematical Biology*, 15:351–357, 1982.

[105] SHARPE. http://people.ee.duke.edu/∼chirel/irisa/sharpegui.html, 2013.

[106] G. Shedler and C. Tung. Locality in page reference strings. *SIAM Journal on Computing*, 1(3):218–241, 1972.

[107] MATLAB Statistics. http://www.mathworks.com/products/statistics, 2013.

[108] Simon Struck, Matthias Gdemann, and Frank Ortmeier. Efficient Optimization of Large Probabilistic Models. *Journal of Systems and Software*, (0), 2013.

[109] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons, 2002.

[110] P. Tzelnic. On LRU Stack Model Suitability. *Communication*, 21(7):593, 1978.

[111] H. Venkataraman and G. Muntean. *Cognitive Radio and its Application for Next Generation Cellular and Wireless Networks*. Springer, 2012.

[112] R.D. Yates and D.J. Goodman. *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. Wiley, 2005.

[113] YMER. http://www.tempastic.org/ymer/, 2013.

[114] G. Zoubin. An Introduction to Hidden Markov Models and Bayesian Networks. *International Journal of Pattern Recognition and Aartificial Intelligence*, 15(1):9–42, 2001.

Biography

Education

- **Concordia University**: Montreal, Quebec, Canada

  Ph.D candidate, Electrical & Computer Engineering, (May. 09 - present)

- **Concordia University**: Montreal, Quebec, Canada

  M.Eng, Electrical & Computer Engineering, (Sep. 06 - Jun. 09)

- **Wuhan University**: Wuhan, HuBei Province, China

  M.A.Sc, Electronics Engineering, (Sep. 96 - Jun. 99)

- **Shaoyang University**: Shaoyang, Hunan Province, China

  Diploma in Industrial Electrical Automation Study, (Sep. 91 - Jun. 94)

Awards

- Outstanding Female Employee in China Space Aeronautics Electro-mechanics Group, Beijing (2001)

- Excellent Graduate Student, Wuhan University (1996-1999)

- Outstanding Student in Hunan Province (1993)

Work History

- **Concordia University**: Montreal, Quebec, Canada

  Research Assistant, Electrical & Computer Engineering (2009 - present)

- **China Space Aeronautics Electro-mechanics Group Co., Ltd.**: Beijing, China

  Team Leader and Senior Engineer, Digital Signal Department (1999 -2005)

- **Wuhan University**: Montreal, Quebec, Canada

  Research Assistant, Electronics Engineering (1996 -1999)

- **Beijing Shougang Company Ltd.**: Beijing, China

  Technician, Steel Rolling Workshop (1994 -1996)

Publications

- **Journal Papers**

  - **Bio-Jr1** L. Liu, O. Hasan, and S. Tahar. Formal Reasoning about Finite-state Discrete-Time Markov Chains in HOL; *Journal of Computer Science and Technology*, Springer, Vol. 28, No. 2, March 2013, pp. 217-231.

  - **Bio-Jr2** L. Liu, O. Hasan, and S. Tahar. Formal Reasoning about Classified Discrete-Time Markov Chains in HOL; *Journal of Applied Logic*, Submitted in May 2013.

- **Conference Papers**

  - **Bio-Cf1** L. Liu, O. Hasan, and S. Tahar. Formalization of Finite-state Discrete-Time Markov Chains in HOL. *In Automated Technology for Verification and Analysis*, volume 6996 of LNCS, pages 90-104. Springer 2011.

  - **Bio-Cf2** L. Liu, V. Aravantinos, O. Hasan, and S. Tahar. Formal Reasoning about Classified Markov Chains in HOL, In *Interactive Theorem Proving*, volume 7998 of LNCS, pages 295-310. Springer 2013.

  - **Bio-Cf3** L. Liu, V. Aravantinos, O. Hasan, and S. Tahar. On the Formal Analysis of HMM using Theorem Proving, Submitted to *Certified Programs and Proofs*, June 2013.

  - **Bio-Cf4** L. Liu, O. Hasan, and S. Tahar. Formal Analysis of Memories Performance, Submitted to *Brazilian Symposium on Formal Methods*, June 2013.

- **Technical Reports**

  - **Bio-Tr1** L. Liu, O. Hasan, and S. Tahar. Formalization of Discrete-Time Markov Chain in HOL, Technical Report, Concordia University, December 2010.

  - **Bio-Tr2** L. Liu, O. Hasan, and S. Tahar. On the Formalization of Continuous-Time Markov Chain in HOL, Technical Report, Concordia University, May 2013.